Information Mining Technologies to Enable Discovery of Actionable Intelligence to Facilitate Maritime Situational Awareness

I-MINE

Marie-Odette St-Hilaire
Melita Hadzagic

Prepared by:

OODA Technologies Inc.
4891 Av. Grosvenor, Montreal, QC, H3W 2M2

Project Manager: Anthony Isenor
Contract Number: Call-up 8, No. 4500959431 to W7707-115137
Contract Scientific Authority: Sean Webb, DRDC Atlantic

DRDC-RDDC-2014-C96

Contract Report

January 2013

Abstract

Human operators trying to establish individual or collective maritime situational awareness often find themselves overloaded by huge amount of information obtained from multiple and possibly dissimilar sources. This kind of situation has also been identified within Maritime Forces Atlantic (MARLANT) and its supporting activities in the Regional Joint Operations Center (RJOC) East and West and the Marine Security Operations Centres (MSOCs) as its current information infrastructure (e.g. Global Position Warehouse (GPW)) faces a challenge of how to extract/discover valuable knowledge from the available large volumes of maritime traffic information usually stored in large databases.

Applying data mining techniques to large sets of maritime traffic data to extract knowledge will facilitate vessel traffic analysis and management for maritime analysts as well as improved decision-making in the maritime domain. Since maritime traffic data differs from the data commonly mined in business domains, the selection of appropriate data mining tools is crucial for meaningful knowledge extraction.

This report provides an extensive review and explores potential use of available information/data mining technologies by maritime analysts to enable discovery of actionable intelligence to facilitate maritime situational awareness. The focus is on open source data mining tools, while the data is restricted to spatio-temporal maritime traffic data such as the Automated Identification System (AIS) data. It includes assessments of selected data mining tools using the scenarios of potential interest to the maritime environment covering both user and administrator perspectives. The report also presents an introductory theoretical background on data mining with special attention to spatial and spatio-temporal data mining as well as an overview of organizations and institutions that work on data mining with data sets similar to maritime traffic data.

This page intentionally left blank.

Table of contents

List of figures

List of tables

1 Introduction

Human operators trying to establish individual or collective maritime situational awareness often find themselves overloaded by a huge amount of information obtained from multiple and possibly dissimilar sources. The requirement of the International Maritime Organization (IMO) to install Automatic Identification System (AIS) on board ships, together with the use of other self-reporting systems based on Global Positioning System (GPS)-quality navigation information [3], contribute to the overabundance of information, which is potentially of great value and importance, but typically though, the resources are not fully exploited. In such circumstances, there is a challenging issue of how to extract/discover valuable knowledge from the available large volumes of maritime traffic information usually stored in a large Database (DB).

The process of Knowledge Discovery in Databases (KDD) has been defined in [1] as an interactive and iterative nontrivial process which includes planning, data integration, selection of target data, data cleaning and pre-processing, data reduction and transformation, selection of suitable data mining techniques to support the discovery process, and evaluation, presentation and interpretation of results, a subset of which may be considered as new *knowledge*. Within the overall KDD process, the data mining is viewed as the sub-process concerned with the discovery of *hidden* information.

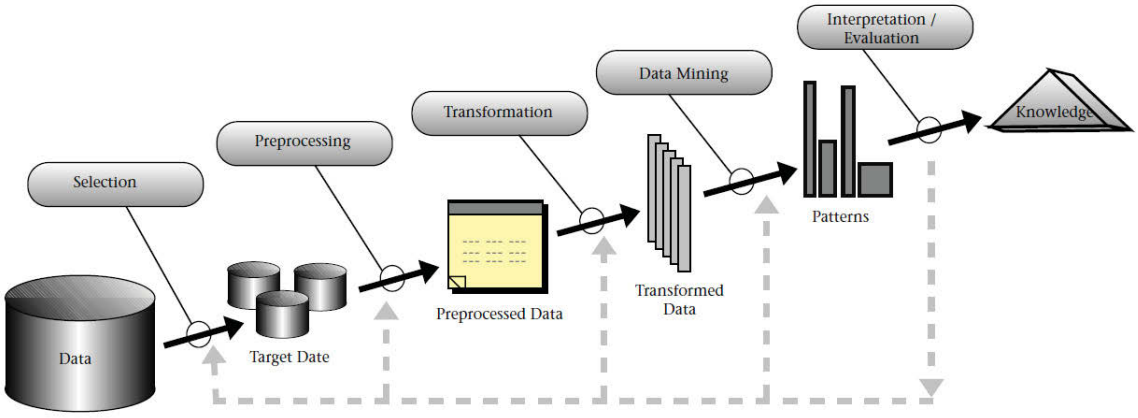

Figure 1: *Steps involved in KDD [1].*

Applying data mining techniques to large sets of marine traffic data to extract knowledge will facilitate vessel traffic analysis and management as well as improved decision-making in the maritime domain. Since maritime traffic data differ from the data commonly mined in business domains (e.g. Data Mining (DM) in Business Intelligence (BI) [1]) [4], the selec-

1. BI is much broader category than data mining. For example, creating a report with monthly sales activity by salesperson and product is called BI. Dashboards and scorecards are also BI tools. Data extraction, transformation, OLAP, slicing, dicing, filtering are all related to BI processes. Data mining is a BI tool, too.

tion of appropriate data mining tools is crucial for meaningful knowledge extraction. The goal of this project is to explore available information/data mining technologies to enable discovery of actionable intelligence to facilitate maritime situational awareness. The focus is on open source data mining tools, while the data is restricted to spatio-temporal maritime traffic data such as the AIS data.

The assessment of the selected data mining tools is performed using data stored in Maritime Situational Awareness Research Infrastructure (MSARI) DB, [5]. MSARI DB, currently under development, is envisaged to support the Defence Research and Development Canada (DRDC) Maritime Information Support (MIS) group in its current and future research efforts related to wide area surveillance in Canada's three oceans. The major functionalities of MSARI DB are: (i) acquiring data from several data sources, such as AIS and Automatic Dependent Surveillance-Broadcast (ADS-B) data sources, (ii) storing and providing means to maintain the data, (iii) providing the capability to add applications for data processing and (iv) query capabilities to access the data. As such, it constitutes an appropriate candidate database for spatio-temporal maritime traffic data mining with a projected size of approximately 5 TB for one continuous year of data.

This document summarizes all the technical activities and findings of the Call-up 8 against the W7707-115137 contract. It provides a review of data mining tools with potential use by maritime analysts, covering both user and administrator perspectives and including expected user expertise and example applications relevant to the maritime environment. Moreover, it presents a detailed assessment of selected data mining software tools (systems) using relevant scenarios and data retrieved from the MSARI DB. Specifically, it includes:

– An introductory theoretical background on data mining with special attention to spatial and spatio-temporal data (e.g. AIS data) mining, (Section 2)
– An overview of organizations and institutions that work on data mining with data sets similar to marine traffic data, including the descriptions of data and tool types they are using, (Section 3),
– The list of relevant data mining software tools/systems and descriptions of selected DM tools, (Section 4)
– The descriptions of data sets and scenarios used to assess the selected DM tools, (Section 5),
– The assessments of the selected DM tools, (Section 6 and Section 7), and
– The summary of the assessments and recommendations for integration with current and future Maritime Forces Atlantic (MARLANT) systems. (Section 8)

2 Data Mining

The most well-known definition of data mining, given by Frawley et al.(1991), in [6], defines data mining as a set of mechanisms and techniques, realized in software, used to extract implicit, previously unknown (or *hidden*) and potentially useful information from large databases. The word hidden in this definition is important; Structured Query Language (SQL) style querying, however sophisticated, is not considered as data mining. In addition, the term information should be interpreted in its widest sense. By the early 1990s, data mining was commonly recognized in computer science as a subprocess within the KDD. In the modern context of data mining, the term Knowledge Discovery in Data would be more apt, as we are no longer preoccupied solely by databases [4]. Nowadays, there are plenty of data mining techniques for tabular data available, which are extensively performed by many commercial enterprises and researchers for mining tabular data, using software such as RapidMiner [2], KNIME [3] or others (see Section 4), on standard desktop machines.

The current focus of the data mining community is the application of data mining to non-standard data sets (i.e. non-tabular data sets) such as image sets, documents, video, multimedia data, network data, matrices, graphs and tensors. For the last three listed data sets, the data mining algorithms employ methods and algorithms from advanced matrix computations, graph theory and optimization [7]. In these methods, the data are described using matrix representations (graphs are represented by their adjacency matrices) while the data mining problem is formulated as an optimization problem with matrix variables. With these, the data mining task becomes a process of minimizing or maximizing a desired objective function of matrix variables. Examples include spectral clustering, non-negative matrix factorization, Principal Component Analysis (PCA) and Singular Value Decomposition (SVD) related clustering and dimension reduction, tensor analysis and L-1 regularization.

The data mining techniques can be seen as a mixture of approaches to machine learning and statistics [4]. There is, however, a distinction between data mining and machine learning. Data mining is focused on data (in all its formats) and as such can be viewed as an application domain while machine learning, at least in its traditional form, is focused on mechanisms whereby computers can learn (e.g. one focus of early work on machine learning was computer programs that could learn to play chess). Machine learning can thus be viewed as a technology, whereas data mining, and by extension KDD as an application.

Most data mining techniques are heuristics tailored to discover patterns of a generic type such as classes, associations, rules, clusters (the ”large patterns”) and outliers (the ”small patterns”) [8]. Since these techniques are heuristics, there is no single optimal algorithm

2. www.rapidminer.com
3. www.knime.org

for discovering patterns of a given type; different techniques highlight different aspects of the information space implied by the database at the expense of other characteristics.

Various pattern discovery techniques can be described more in detail as follows:

(i) Rule pattern extraction/identification
  – Rule pattern (often used as only ”pattern”) recognition has been one of the primary goals of data mining (e.g. identifying purchasing/sales patterns, trends in temporal or longitudinal data, etc.) A pattern is any frequently occurring combination of entities, events, objects, etc. The association rule, as first proposed by Agrawal et al. (1993), [9], in the context of super market basket analysis, is the most known pattern mining method while the most popular current frequent pattern mining algorithm is the frequent pattern growth [10] by Han (2000).

(ii) Data clustering
  – Clustering is grouping of data into categories, a technique also used in machine learning. Coenen (2011), [4], also states that there is no ”best” clustering algorithm applicable to all data; instead, for reasons that are not entirely clear, some algorithms work better on some data sets than others. The K-means algorithm [11] by MacQueen (1967), is one of the most commonly used algorithms if the number of clusters is known. Other clustering algorithms include classifications according to some proximity threshold, such as K-Nearest Neighbor (K-NN) [12] by Hastie and Tibshirani (1996), hierarchical clustering where the data are iteratively partitioned to form a set of clusters, such as BIRCH [13] by Zhang et al. (1996), and model-based clustering where a model is hypothesized for each of the clusters. Assuming that the data are generated by a mixture of underlying distributions, the idea is to optimize the fit between the data and the model, e.g. COBWEB algorithm [14] by Fisher (1987).

(iii) Classification/Categorization
  – Classification involves building *classifiers* of data so as to categorize the data into classes. Unlike clustering, it requires pre-labelled training data from which the classifiers can be built. As such, classification in data mining is sometimes referred to as supervised learning while clustering is considered as unsupervised learning. A typical classification algorithm in application to KDD is a decision tree classifier, such as the one described by Rastogi and Shim (1998) in [15].

(iv) Outlier/Singularity/Anomaly detection/identification
  – This class of data mining algorithms makes possible identifying rare events and exceptional cases in data usually referred to as ”small patterns”. For example, in vessel behaviour analysis, this means identifying anomalies. This usually requires the development of a model representing normal vessel behavior. The anomalous behavior is then identified by the degree to which a vessel’s motion does not conform to that model normalcy. The subject of motion behaviour analysis, however, is not unique to maritime surveillance. It has initially been studied in computer vision in the context of traffic monitoring, human activity monitoring,

and unusual event detection. In the literature, there are various machine learning techniques used to generate normalcy models for analyzing vessel behaviour and helping in the detection of anomalies from e.g. AIS data using Bayesian networks [16], kernel density estimation [2], Gaussian mixture models [17], support vector machines [18], neural networks, etc.

It is important to emphasize that a DM process is interactive, iterative and exploratory. A standard data mining task represents itself as a KDD process as defined in [1]. Specifically, it includes the following steps [19]:

1. **Setting the target:** Understanding the domain in which data is to be mined in terms of clearly describing the objectives, and listing the possible assumptions and anticipated desired results.
2. **Establishing the target data set:** Choosing the initial data set to be analyzed, e.g. AIS data set.
3. **Data pre-processing:** Using effective or readily available approaches to process noisy or incomplete data, e.g. decoding and amending AIS into DB or GIS processing of spatial information.
4. **Data cleaning and transformation:** Deleting or adding some attributes using standardization and/or conversion methods.
5. **Data mining:** Applying most appropriate data mining algorithms to optimally process data by
   - 5.1 *Choosing the data mining task.* This involves selecting the generic type of pattern sought through data mining; this is the language for expressing facts in the database. Generic pattern types include classes, associations, rules, clusters, outliers and trends.
   - 5.2 *Choosing the data mining technique* for discovering patterns of the generic type selected in the previous step. Since data mining algorithms are often heuristics (due to scalability requirements), there are typically several techniques available for a given pattern type, with different techniques concentrating on different properties or possible relationships among the data objects.
   - 5.3 *Data mining.* Applying the data mining technique to search for interesting patterns.
6. **Explanation and evaluation:** Searching for useful and interesting information. If none, repeat the previous steps with other attributes and samples.
7. **Action:** If mined knowledge is found useful then it is integrated and applied to solve the appropriate problem, supporting decision making.

Each step requires expertise from a domain expert, a data analyst and a data miner. The results from data mining are usually presented in the form of concepts, rules, regularities, patterns, constraints and visualization.

2.1 Spatial Data Mining

Spatial Data Mining (SDM) is the process of discovering interesting and previously unknown, but potentially useful patterns from spatial databases [20].

The data inputs of spatial data mining have two distinct types of attributes: non-spatial attributes and spatial attributes. Non-spatial attributes are used to characterize non-spatial features of objects (e.g. name, population, ID, and unemployment rate for a city). They are the same as the attributes used in the data inputs of classical data mining. Spatial attributes are used to define the spatial location and extent of spatial objects[21]. The spatial attributes of a spatial object most often include information related to spatial locations (e.g., longitude, latitude and elevation, as well as shape). Relationships among non-spatial objects are explicit in data inputs, e.g., arithmetic relation, ordering, is instance-of, subclass-of, and membership-of. In contrast, relationships among spatial objects are often implicit, such as overlap, intersect, and behind.

Specific features of spatial data that preclude the use of general purpose data mining algorithms are:

(i) the spatial relationships among the variables; SDM can be characterized by Tobler's first law of geography [22] which states that near things are more related than distant things.
(ii) the spatial structure of errors;
(iii) the presence of mixed distributions as opposed to commonly assumed normal distributions;
(iv) observations that are not independent and identically distributed (i.i.d.);
(v) spatial auto correlation among the features, which captures the property (iv) and augments standard DM techniques for SDM, and
(vi) nonlinear interactions in feature space.

The complexity of spatial data and implicit spatial relationships limit the usefulness of conventional data mining techniques for extracting spatial patterns. Although conventional data mining algorithms can be applied under assumptions such as i.i.d., these algorithms often perform poorly on spatial data due to their self-correlated nature [20]. In [20], Shekhar et al. (2011), provide a list of references to various spatial data mining algorithms and applications in domains such as public health, mapping and analysis for public safety, transportation, environmental science and management, economics, climatology, public policy, Earth science, market research and analytics, public utilities and distribution, etc.

A subset of SDM are geographic data mining algorithms. While SDM deals with physical space in general, from molecular to astronomical level, geographic data is data related to the Earth. Almost all geographic data mining algorithms can work in a general spatial setting (with the same dimensionality).

For non-spatial DB, it is estimated that between 60% and 80% of the time and effort in the KDD process is dedicated to data preparation. This problem increases significantly for geographic databases because of the greater complexity of geographic data [23].

Pattern types such as classes, associations, rules, clusters, outliers and trends all have spatial expressions since these patterns can be conditioned by the morphology as well as spatial relationships among these objects. References to current state-of-the-art algorithms in spatial/geographical data mining, especially in the areas of prediction and classification, outlier detection, spatial co-location rules, and clustering techniques can be found in [21].

2.2 Spatio-Temporal Data Mining

Like spatial data, which requires consideration of spatial auto correlation and spatial relationships and constraints in the model building, spatio-temporal data mining also requires explicit or implicit modeling of spatio-temporal auto correlation and constraints [24]. In [25] several spatio-temporal extensions of classification, clustering, and outlier detection are mentioned.

Modeling spatio-temporal data requires keeping track of spatial location, $s \in D$ at time point t. Different data types are possible in regards to which points s are observed in D. The set of spatial locations D can be fixed (e.g. air pollution data), or can vary with time (e.g. data obtained from a ship measuring ocean characteristics as it moves). Hence, two important data types are commonly considered:

1. *spatio-temporal aerial (or block level) data* - when the fixed region D is partitioned into a finite number of aerial units with well defined boundaries (e.g. postal codes, districts, etc.). Here, an observation is thought to be associated with an aerial unit of non-zero volume rather than a particular location point. Typical aerial data are represented by a choropleth map which uses shades of color or gray scale to classify values into a few broad classes. Such a map provides adjacency information of the aerial units (blocks or regions). Spatio-temporal smoothing, inference and predictions for new aerial units are some commonly used statistical techniques for processing this kind of data type.
2. *spatio-temporal point data* - when D is random region in which an event of interest occurs (e.g. outbreak of a disease). i.e. D is indexed by the spatial point. There is one important practical distinction to be made between the processes defined as discrete-time sequence of spatial point process, and spatially and temporally continuous point process. The latter case typically refers to trajectories of moving objects over time, which consist of sampled locations at specific timestamps.

One of the frequent problems in spatio-temporal data mining with many applications such as identifying tactics in battlefields, games, and predator-prey interactions is the problem of spatio-temporal co-occurrence pattern mining (e.g. Mixed-Drove Spatio-temporal Co-Occurrence Patterns (MDCOPs)). MDCOPs represent subsets of two or more different

object-types whose instances are often located in spatial and temporal proximity. Mining MDCOPs is computationally very expensive because the interest measures are computationally complex, datasets are larger due to the archival history, and the set of candidate patterns is exponential in the number of object-types. Celik et al, 2008 presented in [26] a monotonic composite interest measure for discovering MDCOPs and a novel MDCOPs mining algorithm where the aim was to discover MDCOPs representing subsets of different object-types whose instances are located close together in geographic space for a significant fraction of time. Unlike the objectives of some other spatio-temporal co-occurrence pattern identification approaches where the pattern is the primary interest, in MDCOPs both the pattern and the nature of the different object-types are of interest.

2.2.1 AIS Data Mining

AIS data obtained from ship/airplane onboard AIS transponders have an important role in littoral state monitoring. As the most important self-reporting maritime system, which has been made compulsory by the IMO for most commercial ships, its use can be extended to sharing data between VTS and national administration, gathering information on the presence and patterns of traffic, planning aids to navigation, legal evidence and accident investigation, search and rescue, risk analysis and generating statistics. AIS messages are automatically broadcast with a reporting frequency directly proportional to the speed of the vessel.

AIS information include:

– ship static information, programmed into the unit at commissioning
– voyage related data, entered manually by the master through a password protected routine, and
– dynamic positioning data, derived from interfaces with the ship's GPS and other sensors.

Some efficient and recent AIS data mining techniques are summarized below.

The most important AIS data mining technique involves extraction and definition of motion patterns. In the case of detecting anomalies in ship motion, an anomaly detection algorithm is subsequently applied. Definition of motion patterns can be quite complex if one deals with a network of origins/destinations with multiple connecting paths. See for example Figure 2, which illustrates a network of vessel paths in a harbour, where node 1 may be the entry or the exit point of a harbour, while nodes 2, 3 and 4 are the docking stations in the harbour.

Assuming that the motion trajectories have been already extracted, Ristić et al. (2008) in [2] define a motion pattern by kinematic and attribute information, with only one compulsory attribute - its origin. Other useful attributes, if available, can be the vessel type, season of the year, etc. The kinematic information includes the ship location (in two-dimensions)

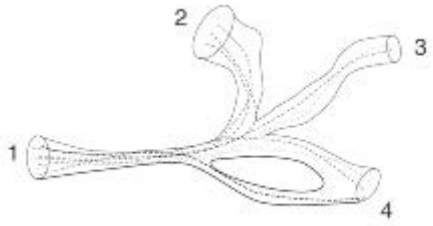

Figure 2: *A network of vessel paths. Example from [2].*

and velocity (also in two-dimensions), while the origin of a motion pattern is defined by the location-velocity vector and its associated uncertainty ellipsoid. Ristić et al. also suggest that it is very useful for a pattern to contain the elapsed time information in the form of the interval of time since the vessel was at the origin of the pattern because the complexity about the topology of the network of paths is then eliminated. However, the authors do not provide information on how the motion patterns have been extracted.

AIS reports are timestamped, but some of traditional data mining techniques lose the time stamp and represent a navigation trajectory as a set, rather than a sequence, of consecutive latitude-longitude vessel positions. To include the timestamp information in the process of maritime traffic modeling, Bruno and Appice (2011) in [27] applied a multi-relational data mining method called SPADA where relational patterns (i.e. patterns which may involve several relations at once) and association rules are discovered from a relational DB in which data are stored. The vessel data and AIS data are modeled as distinct relational data tables (one for each data type) which helps distinguishing between the reference objects of analysis (vessel data) and the task-relevant objects (AIS data), and representing their interactions. The modeled interactions also include the total temporal order over AIS reports for the same vessel and interesting associations between a vessel (reference objects) and a navigation trajectory. Each navigation trajectory represents a spatio-temporal pattern obtained by tracing the subsequent AIS reports (task-relevant objects) of vessels.

In [28] Oo et al. (2004) present a fuzzy inference based model for identifying congested zones by investigating the current varying maritime traffic speed. The model is based on the improved Density-Based Spatial Clustering of Applications with Noise (DBSCAN) [8] algorithm by using the neighborhood three models.

In [16] Mascaro et al. (2010), after the pre-processing that involves cleaning and separating the AIS data into tracks, applied the machine learner CaMML on AIS data combined with additional real world data, and used this to produce two networks time scales, in the form of the time series and track summary models. By adding some real world attributes the normalcy model improved, however, weather variables proved to have no impact on vessel behaviour.

In [29], a DM platform based on AIS data (ADMP) has been proposed by Tang and

Shao (2009), and which produces the eigenvalue of marine traffic using clustering and statistics. The ADMP offers basic data support for data mining, marine traffic flow forecast and development and programming of marine traffic engineering.

In [30], Zhu F. (2011), applies Agrawal's association rule [9] in mining ship trajectory patterns.

Another approach to mining vessel traffic flow data has been proposed in [31] by Zheng et al. (2008). It uses the K-Means clustering algorithm [11] from Waikato Environment for Knowledge Analysis (WEKA) data mining tool [4] to extract multi-factor related regulations according to which clusters were generated. The considered factors include hour, direction, tonnage and ship type.

Tsou (2010) in [32], presented a framework similar to BI discovery, and applied data mining techniques used in that framework to the processing and analysis of AIS data. The difference between the two applications is that information received by AIS includes spatial and temporal features. In order that these features can be distinguished and the relevant knowledge extracted, the AIS received data are first decoded and converted to a readable format. The database management system is used for storage and management, while the GIS analyses and processes the information, converting text data to meaningful spatial and temporal data. This data is then warehoused, while the GIS and BI analysis tools are subsequently used separately to perform visual, spatial and temporal data mining to discover the status and regularities of maritime traffic flow. ArcGIS is used to perform visualization data mining, and SQL Server 2005 Business Intelligence module's association rule mining and sequential pattern mining methods to perform analysis. The association rules for interpreting AIS data as retail market data can be found in [32].

Although not a typical data mining technique, it is worth mentioning here a query based approach based on ESRI ArcGIS technology to analyzing AIS data for improved maritime awareness, as proposed by Ou and Zhu (2008) in [33]. The relevant statistics are generated querying the AIS DB which consists of a main target table containing set of attributes collected by AIS and three look-up tables of interest.

4. www.weka.net.nz

3 Organizations and Academic Institutions that Use Spatio and Spatio-Temporal Data Mining

This Section provides the overview of organizations and institutions that work on data mining with spatio and spatio-temporal data sets. Most of their data mining techniques are in-house developed applications, often devised in collaboration with academia.

3.1 Organizations

Efficient tools for extracting information from geo-spatial data are crucial to organizations which make decisions based on large spatial data sets. These organizations are spread across many application domains including ecology and environmental management, public safety, transportation, Earth science, epidemiology, and climatology. Some of the most important are:

- **NASA**
  Different data mining techniques are performed across various groups in National Aeronautics and Space Administration (NASA). NASA Earth Observing System (EOS) group performs data mining on Earth science data.

  Information on NASA data mining algorithms for application to commercial aviation data and other large data repositories can be found in [34]. The same source contains links to open source algorithms, which include the Multiple Kernel Anomaly Detection Algorithm (MKAD), and the Multivariate Time Series (MTS) Search [34].

  The Data Mining and Complex Adaptive Systems Group in the NASA ARC Computational Sciences develops data mining techniques in application to the Integrated Systems Health Management (ISHM). Two main applications are:
  – *Data Mining for Fault Detection*: The methods are designed to automatically detect unusual or anomalous data in either historical or real-time sensor data, so that people can direct their attention to the unusual data. The research includes both supervised (using examples of faults) and unsupervised (using only nominal training data) approaches. These methods can also be used to help construct monitors for use with a model-based diagnosis system such as Livingstone.
  – *Data-Driven Modelling for Prognostics*: This approach involves the prognosis of future failure states and predicting the type of failure which is extremely difficult due to the high dimensionality of the problem. The number of relevant dimensions for prognosis of spacecraft failures is in the tens of thousands. There are several scientific approaches to prognostics including data-driven, physics-based, and statistical approaches. A variety of advanced real-time data mining techniques that incorporate

model-based information with sensor data to identify potential precursors of failure are used to forecast trends and potentially anomalous behaviour based on real-time information.

- **NOAA and NODC**
  National Oceanic and Atmospheric Administration (NOAA) is a scientific agency focused on the conditions of the oceans and the atmosphere. They provide environmental information products, related to state of the oceans and the atmosphere, to several US agencies and international organizations. They also perform applied research related to ecosystems, climate, weather and water, and commerce and transportation. As a result of these efforts, they developed several tools to detect patterns and predict oceanographic and atmospheric phenomena and trends.

  Most notable is the Multipurpose Marine Cadastre (MMC) [5] system, an integrated marine information system, that provides various regularly updated ocean related information, particularly useful to those looking to assess suitablity for ocean uses. It has three primary focus areas: Web map viewers and ocean planning tools, spatial data registry and technical support and regional capacity building. A significant part of the MarineCadastre.gov project is the AIS Data Handler and accompanying databases [6] which allow for AIS data parsing and visualization. To use the handler and its add-ons the full ArcInfo license for ArcGIS 10.0 is required for relationship class functionality as well as Spatial Analyst for density grid functionality. A collaboration with academia resulted in several behavioral pattern recognition algorithms (e.g. by Ristic et al. (2008), described in Section 2.2.1 while others can be found on the project website). Data mining tools are in-house code developed in Python, C++ and Java.

- **NGA**
  The National Geospatial-Intelligence Agency (NGA) performs data mining through its GEOINT products, specifically, the Maritime Safety Products and Services, which collects, evaluates and compiles worldwide marine navigation products and databases. It is responsible for maritime safety and hydrographic activities including support to the worldwide portfolio of NGA and NOAA standard nautical charts and hardcopy and digital publications. Electronic access to databases and products is provided at [35].

3.2 Academia

Various academic institutions perform research and development in the area of data mining, both algorithms and frameworks. The projects with special attention to spatio and spatio-temporal data include:

5. http://www.marinecadastre.gov/
6. http://marinecadastre.gov/AIS/default.aspx

- **MALEF**
  A novel framework MultiAgent Learning Framework (MALEF) is developed by a group of researches from the Czech Technical University and the University of Edinbourgh [36]. It is designed for both agent-based distributed machine learning as well as for data mining. The framework is based on the exchange of meta-level descriptions of individual learning process and online reasoning about learning success and learning progress.

- **Mining AIS Data for Improved Vessel Trip Analysis Capabilities**
  The goal of this research project of The National Center for Freight and Infrastructure Research and Education (CFIRE) researchers at the University of Toledo and Vanderbilt University is to demonstrate the feasibility of using shore-based AIS receivers to archive data on vessel movements.
  It concerns developing a methodology for processing AIS data from multiple sites in near real-time as well as developing a capability to support ad-hoc data queries. Such analyses can identify high-risk locations, generate better travel time estimates, detect vessel arrivals, identify key traffic areas for investment and enhancement, and in general lead to a better understanding of vessel traffic within a given area. The project involves the integration of relational database management systems (RDBMS), Geographic Information System (GIS) and the development of custom software routines.

4 Software Overview and Selection

This Section describes mature data mining software (systems/platforms/tools) that are ready to use. It focuses on general data mining capabilities and the potential of use for maritime traffic data mining. It also includes description of some BI platforms as they have become increasingly popular and can sometimes be used for data mining. A comparison, based on several criteria, including popularity is also presented. Finally, two software are proposed for this investigation.

4.1 Data Mining Software

The methodology used to build the list of data mining software to investigate is summarized here:

1. List all open source data mining software. The already compiled list by Mikut and Reischl in 2011 [37] was found to be the most comprehensive and was used for this purpose.
2. Filter the software to get candidates for the investigation. The following software were discarded:
   (a) Software without a Graphical User Interface (GUI);
   (b) Software customized to narrow application fields such as text mining (e.g. UIMA, GATE), image (e.g. ImageJ), graph mining (e.g. pegasus) or gene mining (MEGA);
   (c) Software showing no activity on their web site since 2010 (e.g. ROSETTA);
   (d) Software implementing only one specific family of methods such as artificial neural network;
   (e) Matlab toolboxes, because they depend on Matlab (e.g. Gait-CAD, PRTools);
   (f) Unstable prototypes.
3. Install the software.
4. Explore functionalities using an excerpt of the MSARI DB: a Comma-Separated Values (CSV) file of 75 000 entries with Maritime Mobile Service Identity (MMSI), time stamp, latitude, longitude and other attributes such as speed and course.

The remaining software of interest are described in this section. General findings from installation, testing and documentation are summarized in Tables 1 and 2. In Table 1, the Operating System (OS) *W* refers to Windows, *M* to Mac and *L* to Linux. For SQL DB access, the mention *relational DB with JDBC* means that the software can be connected and work directly on data from SQL databases such as PostgreSQL (with PostGIS), MySQL, Microsoft SQL Server, SQLite, etc. using the appropriate JDBC driver.

The documentation and ease to learn are evaluated based on a 3-point scale, 3 being the highest score. Note that the ease to learn characteristic assumes that the user is familiar with data mining and knows what kind of technique is required to analyze the data. In that context, *easy-to-learn* refers to the facility to learn how to use and exploit the data mining capabilities offered by the software.

Table 1: *General Information*

Tool	**License**	**Language**	**Version**	**API**	**OS**	**SQL DB Access**
RapidMiner	GPL 3 and enterprise edition	Java, XML	5.2, 2012	Java	W, L, M	relational DB with JDBC
WEKA	GPL 2	Java	3.6, 2012	Java	W, L, M	relational DB with JDBC
Orange	GPL 3	C++ and Python	2.6, 2012	Python	W, L, M	No
R/Rattle	GPL 2,3	R	2.15, 2012	R	W, L, M	relational DB with different packages
KNIME	GPL 3 and enterprise edition	Java	2.6, 2012	Java	W, L, M	relational DB with JDBC
Kepler	BSD 3	Java	2.3, 2012	Java	W, L, M	relational DB with JDBC
TANAGRA	custom	C++	1.4, 2012	C++	W	No

Table 2: *Data access and mining capabilities*

Tool	Spatio-temporal mining	Text mining	Easy-to-learn	Documentation
RapidMiner	No	Yes	1	2
WEKA	WEKA-GDPM for pre-processing	Limited	2	3
Orange	No	Add-on Orange-Text	3	3
R/Rattle	Several spatial packages	`tm` package	3	2
KNIME	No	Yes	3	3
Kepler	No	Some prototypes	1	3
TANAGRA	No	No	1	1

4.1.1 RapidMiner

RapidMiner [7] (formerly Yet Another Learning Environment (YALE)) is an open source data mining tool written in Java that is able to perform various types of regressions, classifications, and other data mining tasks. There is also a commercial version which includes the open-source functionalities, additional features, services and guarantees. Both versions of RapidMiner are maintained by Rapid-I.

RapidMiner has more than 400 data mining operators, which also include third party packages, that can be used and combined, following the concept of rapid prototyping, to produce process flows or pipelines. The setup is described by eXtensible Markup Language (XML) files which can be created with a graphical user interface. The XML based scripting language turns RapidMiner into an Integrated Development Environment (IDE) for machine learning and data mining. Furthermore, RapidMiner can be used as a Java data mining library. Links to all source code and binaries can be found on the RapidMiner website.

Main RapidMiner operators can be grouped as:

– *Input/Output:* Operators for data in/out in different file formats including known data mining and learning scheme formats (Attribute-Relation File Format (ARFF), C4.5, ...), CSV, Excel files, data sets from databases (Oracle, MySQL, PostgreSQL, Microsoft SQL Server, Sybase ...) and many more.
– *Pre-processing:* Operators to be applied before the learning process including discretization, example and feature filtering, normalization, sampling, dimensionality reduction, missing and infinite value replenishment and removal of useless features.
– *Feature operators:* Feature selection, weighting and relevance, feature construction and extraction from time series.
– *Learning:* More than 100 learning schemes for regression, classification and clustering tasks.
– *WEKA:* All learning schemes and attribute evaluators of the WEKA learning environment are also available and can be used like all other RapidMiner operators.
– *Performance evaluation:* Validation and evaluation schemes to estimate the performance of learning or pre-processing on data set, e.g. cross-validation, training and test set splitting, leave-one-out, significance tests, large number performance criteria for classification and regression.
– *Meta operators:* Optimization operators for experiment design, such as parameter optimization, learning curves, experiment loops and iterations
– *Visualization:* Logging and presenting results including the visualization of Online 1D, 2D and 3D plots of data and experiment results, built-in color, histogram, and distribution plots, quartile/box plots, high-dimensional data, Support Vector Machine (SVM) functions, Receiver Operating Characteristic (ROC) plots and lift charts.

7. www.rapidminer.com

The documentation is available online, mostly found on RapidMiner website, including the email and users' forum support. It provides an in-depth introduction to data structures and basic terminology in data mining as well as the description of the window based GUI. It also includes a few simple examples of creating a process using available operators and producing results. Rapid-I also offers courses and seminars at various costs which make it possible to get started faster.

Although a large document, the RapidMiner user manual offers limited aid to a beginner user of RapidMiner with a background in data mining. It makes RapidMiner's learning curve quite steep.

Listed here are some of the online resources which have been found useful:

1. http://rapid-i.com/content/view/306/233/lang,en/
2. http://www.meta-guide.com/home/knowledgebase/best-rapidminer-videos
3. http://fr.slideshare.net/dataminingtools/mining-stream-time-series-and-sequence-data
4. http://applieddatamining.blogspot.ca/
5. http://www.youtube.com/watch?v=utKJzXc1Cow

The evaluation of RapidMiner in application to maritime traffic data mining can be found in Section 6.

Both open source and commercial versions of RapidMiner can be integrated with all types of SQL servers (Oracle, MySQL, Microsoft(MS) SQL). The details on the integration through Rapidminer GUI can be found in Section 6. That section also contains several screen capture of RapidMiner.

4.1.2 KNIME

KNIME [8] is a graphical workbench for the entire data analysis process: data access, data transformation, analytics and data mining, visualization and reporting. The open integration platform provides over 1000 modules (called nodes, as opposed to operators for RapidMiner). There exist several extensions to KNIME, allowing, for instance, to integrate R software environment and WEKA. It can be integrated with all types of SQL servers (Oracle, MySQL, Microsoft(MS) SQL).

KNIME is very similar to RapidMiner. Someone comfortable with RapidMiner will also be with KNIME and vice-versa. However, KNIME offers better on-line support, which is very helpful for new users thus making it easier to learn.

Figure 3 shows a screen capture of KNIME graphical workbench.

8. http://www.knime.org/

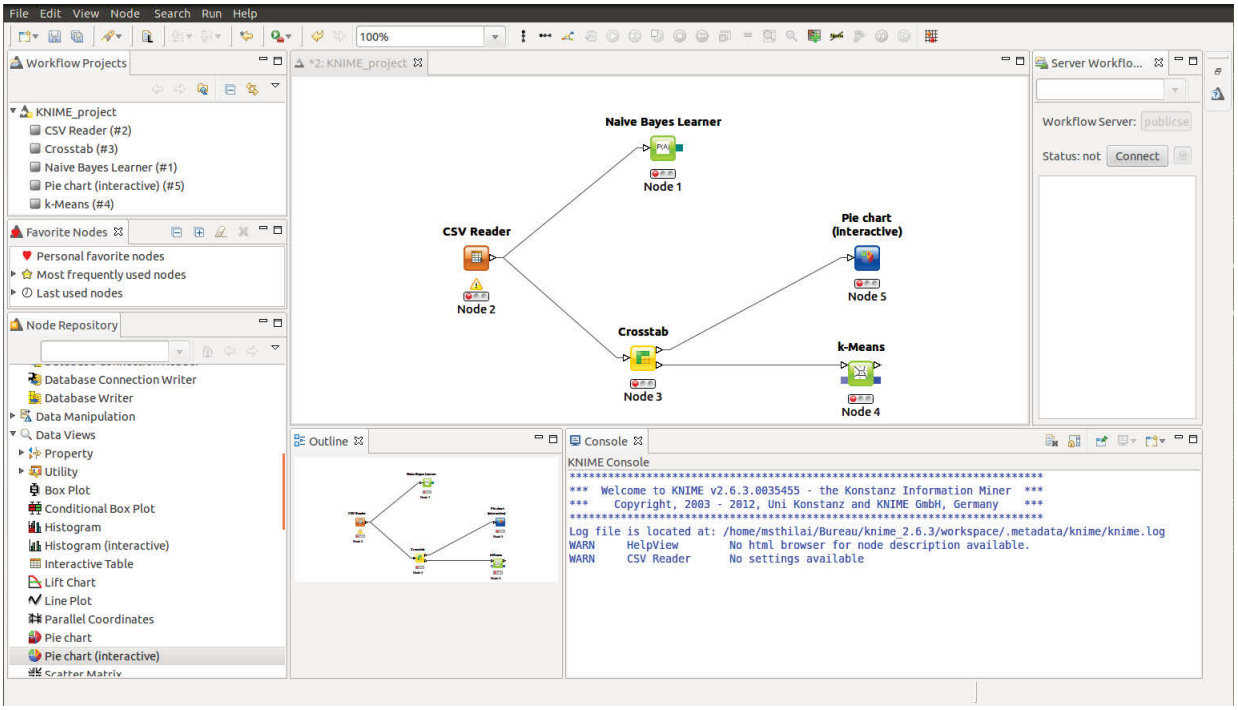

Figure 3: *KNIME screen capture*

4.1.3 Orange

Orange [9] is a component-based data mining and machine learning software suite that features visual programming front-end for exploratory data analysis and visualization, and Python bindings and libraries for scripting. It contains a set of components for data preprocessing, feature scoring and filtering, modelling, model evaluation, and exploration techniques. It is written in C++ and Python, and its graphical user interface is based on cross-platform Qt framework.

Orange supports C4.5, Assistant, Retis, and CSV data formats. It does not allow users to get data from SQL databases. Therefore a user who is interested in mining a database will have to export the data set from the database to use it with Orange.

Orange was found to be the most user-friendly software. It is intuitive and the look-and-feel is great (see Figure 4 for a screen capture). However, it crashes when dealing with larger data sets [10] which makes it not suitable for large amount of data and thus for maritime traffic data mining.

9. `http://orange.biolab.si/`
10. A file of 8 MB was used to test the application.

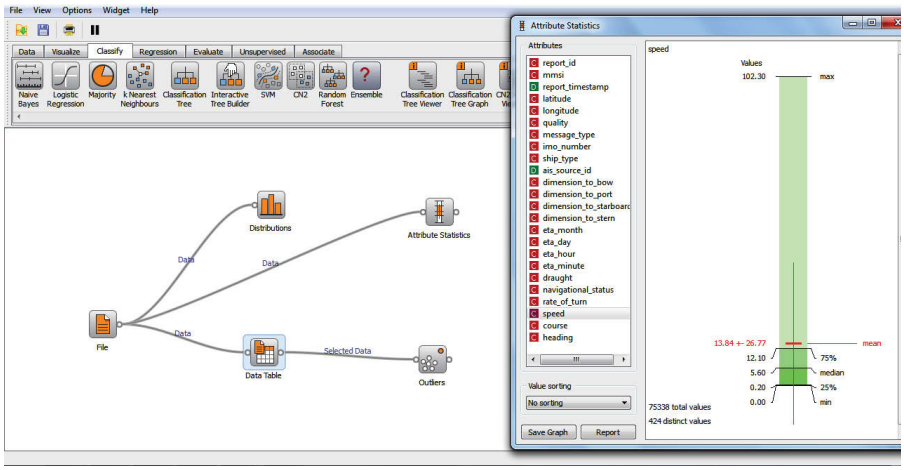

Figure 4: *Orange screen capture: distribution of speed for an AIS reports.*

4.1.4 WEKA

WEKA [11] is a collection of machine learning algorithms for data mining tasks. It implements more than twenty different algorithms for classification, clustering and association rules. The algorithms can either be applied directly to a data set or invoked with the Java Application Programming Interface (API). WEKA contains tools for data pre-processing, classification, regression, clustering, association rules, and visualization. It is also well-suited for developing new machine learning schemes. It can be connected to any relational DB with Java Database Connectivity (JDBC). It also supports ARFF [12] and CSV file formats.

WEKA has a large community of users, which includes machine learning researchers and industrial scientists, but it is also widely used for teaching. WEKA's developers show that they are well aware that the learning curve for data mining is particularly steep by offering exceptional documentation. It offers numerous tutorials, a text book and a very active mailing list for support.

Note that Pentaho (see section 4.2.1) acquired WEKA in 2006.

4.1.4.1 WEKA-GDPM

WEKA-GDPM [13] is an extended version of WEKA to support automatic geographic data pre-processing for spatial data mining. As mentioned in section 2.1, data preparation for geographic data is time and effort consuming. The goal of this tool is thus to simplify that

11. `http://www.cs.waikato.ac.nz/ml/weka/`
12. ARFF files were developed by the Machine Learning Project at the Department of Computer Science of The University of Waikato for use with WEKA.
13. http://www.inf.ufrgs.br/ vbogorny/software.html

step and ultimately save time. Geographic Data Pre-processing Module (GDPM) offers support for PostGIS, for both distance and topological spatial relationships. It allows the user to gather and arrange data stored in a PostgreSQL database with PostGIS extension without SQL scripting. For instance, it is possible to create a data set from the intersection of two geographical regions. That tool can be viewed as a graphical user interface to PostGIS which outputs data sets in the ARFF format, ready to use for WEKA processing. A user comfortable with SQL and PostGIS syntax will feel that this tool is useless.

4.1.5 R

R [14] is an open source programming language and software environment for statistical computing and graphics. It compiles and runs on Linux, Unix, Windows and MacOS. R is widely used by statisticians in industry and academia. There are many packages allowing it to interact with SQL databases and to load different kinds of file formats.

Although R is a statistical language and software environment, it is increasingly used for data mining. See section 4.3 for evidence from survey results.

It is also worth mentioning that the RWeka package [15] provides access from within R to all WEKA data mining algorithms.

4.1.5.1 Rattle

Rattle [16] is a package providing a GUI for data mining using the R statistical programming language. An understanding of R is not required in order to use Rattle. Rattle is simple to use and allows user to rapidly work through the data processing, modelling, and evaluation phases of a data mining project. On the other hand, R provides a very powerful language for performing data mining well beyond the limitations that must be embodied in any graphical user interface and the consequentially canned approaches to data mining. So when there is a need to fine tune and further develop a data mining processes it is possible to migrate from Rattle to R.

Figure 5 is a screen capture of Rattle used to visualize speed and course distributions for different space-based AIS sources.

4.1.5.2 Spatial Data

There is a total of 106 R spatio-temporal analysis packages (see [38]). Among them, the following seem promising for maritime traffic data mining:

14. http://www.r-project.org
15. http://cran.r-project.org/web/packages/RWeka/index.html
16. http://rattle.togaware.com/

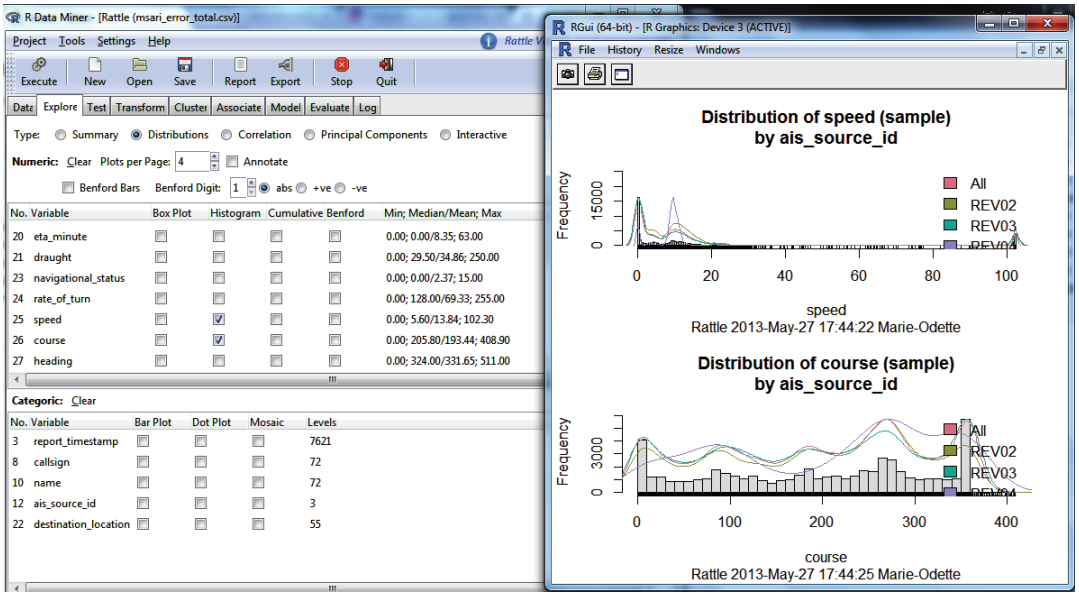

Figure 5: *Rattle screen capture: distribution of speed and course for an AIS reports.*

sp [17] provides a generic set of functions, classes and methods for handling spatial data;

spatstat [18] allows statistical analysis of spatial point patterns;

spacetime [19] is a package with classes and methods for spatio-temporal data.

spatial [20]provides functions for kriging and point pattern analysis.

4.1.6 Kepler

Kepler [21] is designed to create, execute, and share models and analyses. Kepler can operate on data stored locally and over the Web, in a variety of formats (such as data accessible by JDBC, CSV, and also other domain specific format such as Ecological Metadata Language (EML), DiGIR protocol, OPeNDAP protocol, DataTurbine, GridFTP and others). It is an environment for integrating disparate software components. For instance, it can be used to merge R scripts with compiled C code, or facilitating remote, distributed execution of models.

Similarly to RapidMiner, KNIME and Orange, with the GUI users can select and connect pertinent analytical components and data sources, called *actors*, to create an executable representation of the steps required to generate results.

17. `http://cran.r-project.org/web/packages/sp/index.html`
18. `http://www.spatstat.org/`
19. `http://cran.r-project.org/web/packages/spacetime/index.html`
20. `http://cran.r-project.org/web/packages/spatial/index.html`
21. `https://kepler-project.org`

It is possible to extend Kepler with the WEKA data mining functionalities using Kepler-Weka [22], although this is not a very active project (last update in 2010). So even if Kepler can be used for data mining, it is not its primary purpose. Statistical analysis, which is required for data mining, is provided by R and more sophisticated mathematical capabilities are provided through Matlab. Moreover, visualization capability was found to be weak.

4.1.7 TANAGRA

TANAGRA [23] is a data mining software for academic and research purposes. It proposes several data mining methods from exploratory data analysis, statistical learning, machine learning and databases area. The application is very limited in terms of data access: only text files with tab separators can be imported. It was found that the application is better suited for small and exploratory projects. Also, the community of users is very limited.

4.2 Business Intelligence Software

There exist many definitions of business intelligence and depending on individual training and background, the definition of BI may include [39]:

1. An information technology viewpoint: conducting queries that *slice and dice* the data and producing reports and dashboards, possibly using an OnLine Analytical Processing (OLAP) tool;
2. A statistics viewpoint: employing data mining tools to analyze and explore the deluge of data and uncover unforeseen relationships, and;
3. An operations research and management science viewpoint: developing models that impact the organization's strategy, planning and operations, such as Key Performance Indicators (KPI) and user profiles.

Most BI tools have capabilities within each of these areas, with a focus on the first and last ones. However, data mining capabilities are usually only basic. Also, BI tools are implemented via a client/server application, which serves dashboards, reports and data cubes (from OLAP) to clients. This architecture offers no clear advantage for the context of this investigation, or more broadly for maritime traffic data mining activities. Moreover it makes the installation heavier because it requires, among other software, a web server.

For these two reasons, BI platforms were deemed not specialized enough for this investigation and for maritime traffic data mining. However, BI is a very fast growing topic and the limits between data mining and BI are blurring. Business analysts are becoming more knowledgeable in terms of data mining and will become more demanding about their BI platforms. That is why such platforms should not be definitively discarded for mining activities.

22. http://sourceforge.net/apps/trac/keplerweka/wiki
23. http://eric.univ-lyon2.fr/~ricco/tanagra/en/tanagra.html

The main difference between data mining and the reporting and analysis in BI is that in BI reporting the work is usually done on the historical data while searching for certain desired information, e.g. amount of sales/revenues. In data mining, the system is given some data, from which it attempts to find some unknown pattern by using various data mining algorithms described in Section 2, e.g. purchasing patterns, vendors information, etc.

Most of the BI solutions are commercial, but among them two offer open source solutions: Pentaho and Jaspersoft. An exception is SpagoBI, which is a pure open source platform, with capabilities similar to the commercial ones. Both Pentaho and Jaspersoft rely on the same open-source tools: Mondrian (available on SourceForge) and JPivot (also available on Sourceforge). The latter provides the user interface for OLAP , while Mondrian supplies actual OLAP analysis.

Pentaho, Jaspersoft and Spago are compatible with the most popular Relational Database Management System (RDBMS) and they all have strong OLAP, reporting and dashboarding capabilities. For a detailed comparison, refer to Golfarelli's comparison [40].

4.2.1 Pentaho

The community version of Pentaho [24] offers a suite of open source BI products called Pentaho Business Analytics providing data integration, OLAP services (provided by Mondrian server [25]), reporting (partly using the community version of JasperReports), dashboarding and data mining.

Pentaho acquired the WEKA open source data mining project in 2006. Therefore, the data mining capabilities are provided by the WEKA software.

4.2.1.1 GeoMondrian

GeoMondrian [26] is a spatially-enabled version of Pentaho. It is in fact an implementation of the Mondrian OLAP server. It provides a consistent integration of spatial objects, from PostGIS for instance, into the OLAP data cube structure, instead of fetching them from a separate spatial database, web service or GIS file. This tool focuses on the OLAP function of BI, and provides limited statistical and data mining functionalities. One could say that GeoMondrian brings to the Mondrian OLAP server what PostGIS brings to the PostgeSQL DBMS. It provides first spatial extensions to the The MultiDimensional eXpressions (MDX) language, i.e. it adds spatial analysis capabilities to the analytical queries. At present, it only supports PostGIS data warehouses.

24. http://community.pentaho.com/
25. http://mondrian.pentaho.com/
26. http://www.spatialytics.org/projects/geomondrian/

4.2.2 Jaspersoft

The community version of Jaspersoft [27], advertised as Jaspersoft 5.0, is a BI suite mostly known for its reporting and dashboarding capabilities packaged as JasperReports. Jaspersoft also has a data analysis component used to model, manipulate and visualize data using OLAP or in-memory analysis. The main community portal offers several products: *JasperReports Server*, *JasperReports Library*, *Jaspersoft ETL*, *Jaspersoft Studio*, and *iReport Designer*. The product list for the Commercial Edition is much longer, and includes more charting and more data sources. Note that Jaspersoft does not have specific data mining capabilities, such as clustering algorithms for instance. The following describes Jaspersoft open source products.

JasperReports Server is the main server for the reporting engine to which other applications can connect, it is the central product required for the entire suite. It has the option to embed reports, dashboards and analytics into your applications. To do so, *iReport Designer* provides fast design of reports using a graphical designer. Note that iReport Designer is included when you download JasperReports Server which is not immediately clear from the website. *Jaspersoft Studio* is an eclipse-based report designer. *JasperReports Library* is a Java library offering the same kind of reporting capabilities, but can be used from within Java applications. For further support, Eclipse users can produce reports with *Jaspersoft Studio*, an eclipse-based report designer.

JasperReports Server being a web server, it can be accessed through a web browser. There is a menu across the top, along with a list of folders on the left rail of the homepage for managing analysis components, content files, data sources, images, input data types, reports, system properties, and themes. There is also a *Manage* menu for managing users, roles, and server settings. Note that the documentation states that there is a user called *demo* and a *superuser* which proved to be wrong. Instead, username: *joeuser*, password: *joeuser* should be used to access the sample databases.

Reports can be exported in various formats, including PDF, Excel, Excel with pagination, CSV, DOCX, RTF, Flash, ODT and ODS. The various downloaded reports all mostly look the same, whether it was a spreadsheet or word processing document.

The user and role tools are easy to use and work just as you would expect them to. The other item in the *Manage* menu lets you configure various logs, as well as OLAP settings. These are low-level OLAP settings, such as disable memory caching, generate formatted SQL traces, query limit, result limit, and more.

The OLAP tools are an optional component in Jaspersoft BI Suite. Since both Jaspersoft and Pentaho use the same visual interface, JPivot, and the same engine, Mondrian, the following applies to both of them. The OLAP tools let you perform ad-hoc views, reports,

27. `http://community.jaspersoft.com/`

and views through MDX queries. For the ad-hoc views, OLAP searches and analysis can be easily performed inside the web browser. Figure 6 shows one such search: first with products, then drinks, then alcoholic beverages. Meanwhile, the right-hand side updates accordingly, showing the measures (in this case unit sales, store cost, and store sales).

Figure 6: *Jaspersoft OLAP*

Finally, *Jaspersoft ETL*, where ETL stands for Extract, Transform, Load, has capabilities to extract data from a transactional system to create a consolidated data repository for reporting and analysis.

4.2.2.1 Spatially Enabled MDX

There are limited resources available about the descriptions of integrations of geoservers and JasperReports. It is reported in [41] that the Quebec based company Spatialytics [28], whose main product is a lightweight cartographic component which enables navigation in geospatial (Spatial OLAP (SOLAP)) data cubes, such as those handled by GeoMondrian, has made efforts towards integration into existing dashboard frameworks such as Jaspersoft, in order to produce interactive geo-analytical dashboards. Such dashboards aim at supporting the decision making process by including the geospatial dimension in the analysis of enterprise data. One of their earlier version GUI is shown in Figure 7.

4.2.3 SpagoBI

SpagoBI [29] is a collection of BI related open source software, among them: WEKA and R for data mining, Mondrian for OLAP and JasperReports for reporting. But SpagoBI

28. http://www.spatialytics.com
29. `http://www.spagoworld.org/xwiki/bin/view/SpagoBI/`

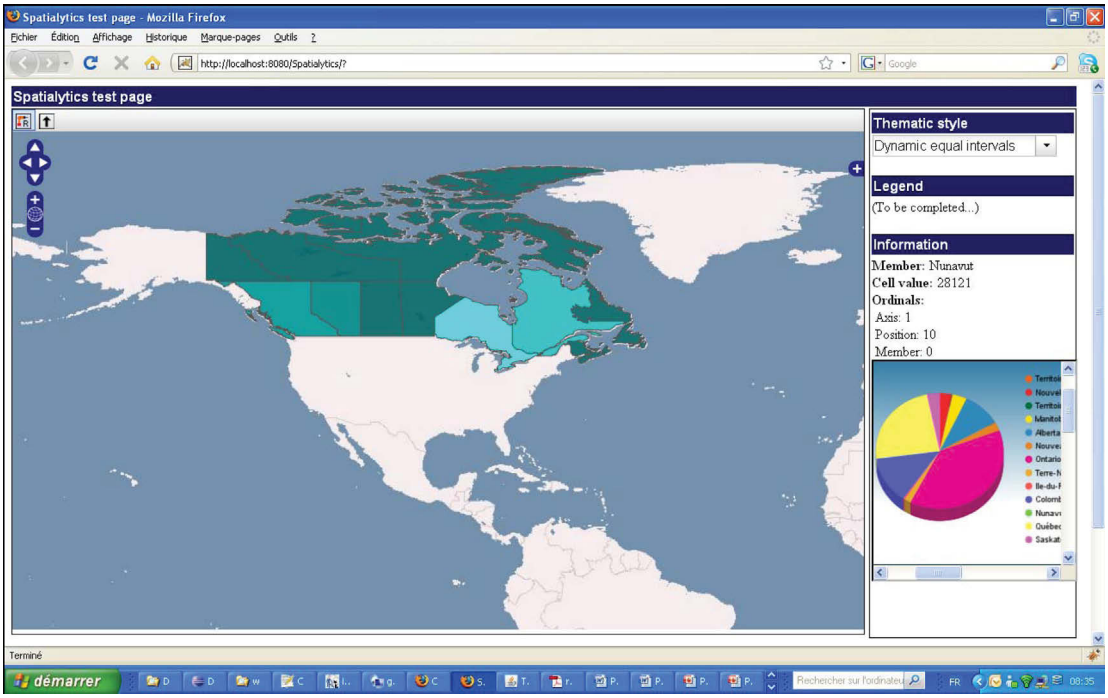

Figure 7: *Spatialytics GUI*

also has its own components, such as geographical engines allowing users to set run-time connections between geographical data and business data stored in the data warehouse. More precisely [42]:

- GEO builds maps showing location-based analysis of patterns and trends.
- GIS engine interacts with real spatial systems, according to the Web Feature Service (WFS)/Web Map Service (WMS) standards.

SpagoBI requires manual configuration at installation, making it cumbersome to setup, at least compared to all other software tested. However, it offers an impressive range of BI functionalities, all for free.

4.3 Popularity

Lately, several surveys have been conducted on data mining tools around the world. Among them, there are the 5th Annual Data Miner Survey from Rexer Analytics in 2011 [43] and the 13th annual KDnuggets Software Poll in 2012 [44].

The Rexer survey has 52 questions, was sent to over 10,000 data miners and had 1,319 respondents from over 60 countries. The data was collected in the first half of 2011.

The KDnuggets poll has 3 questions and was answered by 798 participants, collected in the first half of 2012.

Table 3: *Popularity of data mining tools*

Tool	Liscence	Rexer 2011	KDnuggets 2012	Average Score
R	O	2	1	1.5
RapidMiner	O	4	3	3.5
Statistica	C	1	6	3.5
SAS	C	3	7	5.0
KNIME	O	7	4	5.5
WEKA	O	8	5	6.5
IBM SPSS Statistics	C	5	10	7.5
IBM SPSS Modeler (Clementine)	C	6	11	8.5
Matlab	C	9	9	9.0
Excel	C	-	2	-
TANAGRA	O	-	-	-
Orange	O	-	13	-
Kepler	O	-	-	-

Table 3 summarize the popularity of data mining tools among data miners, from corporate, government and academics. The licence *O* means open source while *C* is for commercial. As for the value provided for each survey, smaller is the value, higher is the popularity and a dash means that the software was not mentioned by the respondents. The average score was computed for this investigation and should be only used as a rough guide to rank software. The results shown in this table are from two slightly different questions:

– 2011 Rexer Survey: What Data Mining software package do you use most frequently as primary tool?
– 2012 KDnuggets poll: What Analytics, Data mining, Big Data software you used in the past 12 months for a real project (not just evaluation)?

Both surveys reported that the proportion of data miners using R is rapidly growing since 2007. Also, it is interesting to see that two of the three most popular data mining software are open source.

4.4 Selection

For this investigation, it was decided to use RapidMiner and R/Rattle.

Since WEKA is already integrated in RapidMiner, R, KNIME and Kepler, it seemed less pertinent to use it individually for the investigation. Also, since KNIME and RapidMiner are found to be very similar, it has been decided to select only one of them to bring diversity in terms of functionalities and usability. RapidMiner has been selected over KNIME because it was a requirement and has also been found to be more popular than KNIME.

R (with Rattle) was selected because of its:

– increasing popularity,
– flexibility: would ease the development of a custom data mining tool for operators,
– large community of users/developers and
– spatio-temporal data analysis capabilities.

In addition, RapidMiner supports R extensibility so new R algorithms can easily be developed or wrapped and then be applied within RapidMiner.

TANAGRA, Kepler and Orange just could not measure up yet with RapidMiner and R.

5 Target Data Sets

This Section describes the data sets used for the scenarios explored with RapidMiner and R.

5.1 Invalid Observations Scenario

The first data set is used for the invalid observations mining scenario, explored with RapidMiner in Section 6.2 and with R in Section 7.2. This scenario investigates detections and relationships between invalid values.

The data used is an AIS data set extracted from MSARI. The data covers the period of November 3 to November 10 2011 and was reported by the exactEarth monitoring system [30]. Data attributes considered for the analysis are: MMSI, latitude, longitude, IMO number, ship dimensions (to bow, port, stern and starboard), ship type, Estimated Time of Arrival (ETA), draught, rate of turn, speed over ground, course over ground and heading. This data set contains a total of 4,321,563 complete reports [31].

The MSARI database has a data quality flag identifying which report has an out-of-range attribute or out-of-range position (see Table 4 for the detailed flags). Therefore, that flag was used to restrict the data to mine. Only reports flagged as having an attribute out-of-range and/or position out-of-range and/or invalid MMSI were selected. The total number of reports flagged with one of these errors is 75,601.

This data set was extracted from MSARI as a CSV file called `msari_error_total.csv`. For the R investigation, the data set was directly extracted from the MSARI DB using R.

5.2 Ship Trajectories Scenario

The second data set is used for the ship trajectories mining scenario, explored with RapidMiner in Section 6.3 and with R in Section 7.3. This scenario focuses on ship movements between ports so the data includes a reports data set and a ports data set.

The reports data set is an AIS data set extracted from MSARI. The data also covers the period of November 3 to November 10 2011 but is limited to the Atlantic coast down to Virginia US, more precisely to the bounding box defined by a latitude between 35 and 53 degrees and longitude between -80 and -48 degrees. This data was also reported by the exactEarth monitoring system. Data attributes considered for the analysis are: MMSI, latitude, longitude, time stamp. This data set contains a total of 93,000 complete reports.

30. `http://www.exactearth.com/`

31. In MSARI, each report corresponds to an AIS message. And a complete report refers to an AIS message that has been successfully parsed.

In order to know if a ship visited a port, it is required to compute distance between the port's location and ship's reported positions. Therefore, this analysis implies the concept of distance. A typical way to deal with that kind of problem is to use a grid covering the region of interest. With that grid, each position (for ports and contacts) can be associated with a cell. If a ship transited to a port, it thus has visited the cell associated to the that port. This is common way to reduce complexity and speed up computation.

The grid and map positions to cells were created using PostGIS. This operation was performed on the MSARI database, with cell width of 0.25 degree, and data was exported as two CSV files ordered by MMSI and time. The file `contact_to_cell.csv` contains the contacts as described above with associated cell IDs and the file `port_to_cell.csv` contains the ports, their position and cell IDs.

The 18 following ports were included in the data set:

Canadian Ports:
– St-John NL,
– Sydney,
– Hawkesbury,
– Halifax,
– St-John NB,
– Sept-Iles,
– Dalhousie,
– Quebec,
– Montreal.

United States (US) Ports:
– Portland,
– Marcus Hook,
– Hampton,
– Baltimore,
– Philadelphia,
– New York,
– New Haven,
– New London,
– Boston.

6 RapidMiner for Maritime Traffic Data Mining

This Section presents a brief introduction to the RapidMiner GUI and descriptions of realizations of several relevant scenarios for maritime traffic data mining. It is assumed that the reader is familiar with basic data structures and data mining terminology.

6.1 RapidMiner Environment

The analysis processes can be produced from a large number of nestable operators and also be represented by so-called operator trees or by a process graph (flow design). The process structure is described internally by XML and developed by means of a GUI. In the background, RapidMiner constantly checks the process currently being developed for syntax conformity and automatically makes suggestions in the case of problems. This is made possible by so-called meta data transformation, which transforms the underlying meta data as early as at the design stage in such a way that the form of the result can already be foreseen and solutions can be identified in the case of unsuitable operator combinations (quick fixes). RapidMiner has the possibility of defining breakpoints, therefore making possible debugging an in-depth analysis of the process by providing insight into the data at specific points in the process (before or after operators of a process). Successful operator combinations can be pooled into building blocks which can be available again in other processes.

The RapidMiner GUI is composed of several windows and views: Repository, Operator, Process, Parameter, Result and Log views. RapidMiner offers a default layout. This layout can be altered through standard functionality provided by the software. RapidMiner allows for straight forward visual process design via a simple drag and drop interface. Operators can be assembled via connections through the operator input and output ports. In simple terms, the process usually consists of a finite number of steps: data load (establishing target dataset or example set which refers to a table of the current examples), data preparation and transformation , mining (applying mining techniques), and learning. Figure 14 displays a prototypical RapidMiner GUI session. For a quick introduction to the RapidMiner GUI, there is a video at `http://rapidminerresources.com/index.php?page=gui-introduction`.

6.2 Mining Invalid Observations

Here, the design of processes of detecting outliers using RapidMiner is described.

We define the data mining steps as:

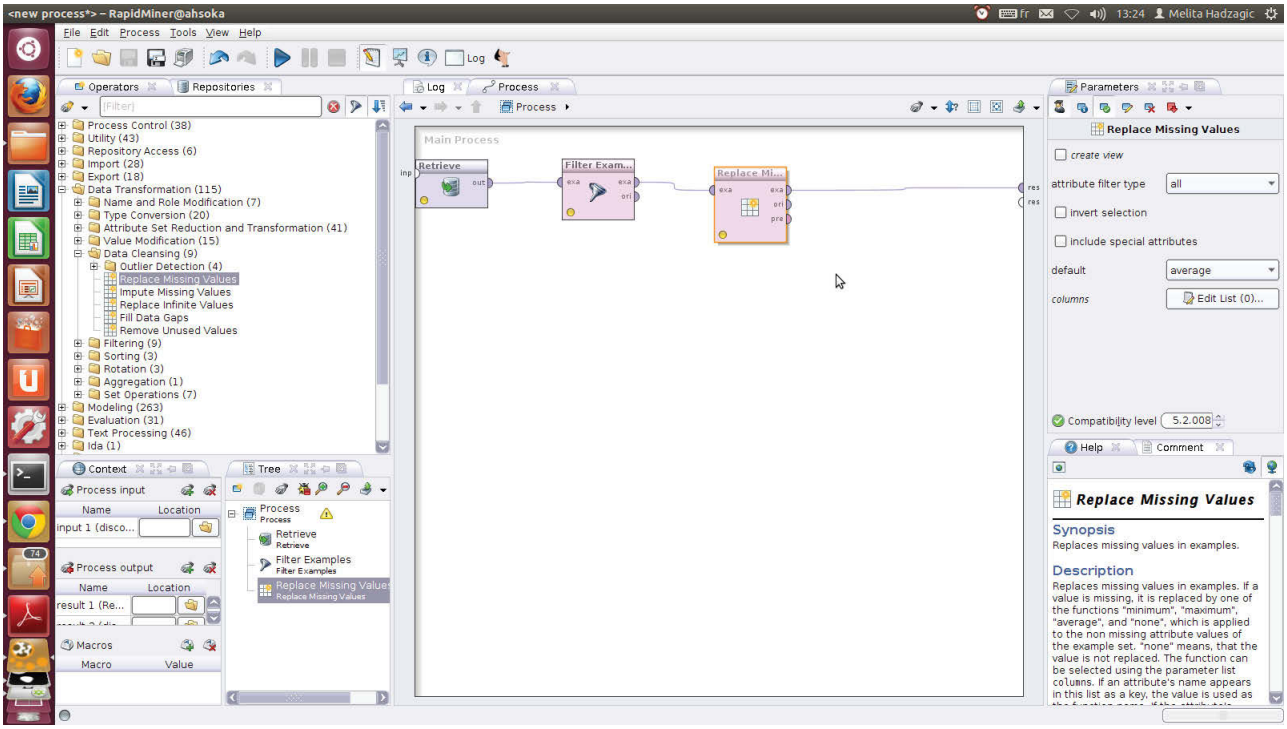

Figure 8: *RapidMiner user interface.*

1. **Setting the target:** The process is to mine the following pattern:
   Detecting out-of-range error in the heading, latitude, longitude and/or speed field.
   where out-of-range errors are defined by the AIS data specifications.
2. **Establishing the target dataset.** The data set is provided in the file **msari_error_total.csv** which has been extracted from MSARI DB. It contains AIS data with the following attributes:

contact ID	MMSI	longitude	latitude	report_timestamp	course	speed	rate OfTurn	...

   The operator *Read CSV* is used to import the data to RapidMiner. The file can be read through the File Browser or through the Import File Wizard. The Import File Wizard also has the options for removing unwanted attributes and verifying/changing data types. Note that when loading the data from a CSV file, either by using Import Wizard or by using the File Browser, RapidMiner may confuse the data types. It is recommended to verify the data types when loading data and change them if necessary. The data types can be changed in *data set meta data information* as well as during the design process. The values and their corresponding meaning for data quality attribute are summarized in Table 4.

3. **Data pre-processing.** First, the duplicates are removed using the *Remove Duplicates* operator, which allows also for removing missing values.

data_quality	Type of error
0	no error
1	Extra characters
2	Position out of range: $lat \notin [-90^0, 91^0]$ or $lon \notin [-180^0, 181^0]$
4	Attribute out of range (as defined by the specs)
8	Position not available ($latitude = 91$ and $longitude = 181$
16	Attribute not available (Default value defined in the source specs)
32	Invalid message format - Data could not be parsed and is stored as raw format only
64	Invalid date/time format - Date could not be parsed and is stored as raw format only
128	Invalid attribute format - Attribute could not be parsed and is stored as raw format only
256	Invalid MMSI format (MMSI could either not be parsed (is stored as raw only) or does not contain 9 digits)

Table 4: *Data quality attribute values and the corresponding error types.*

4. **Data cleaning.** Removing unwanted attributes can be done through Import File Wizard, when one can uncheck the unwanted attributes. Another way of (de)selecting the attributes of interest is by using *Select Attributes* operator or *Filter Examples* operator which allow for selecting a single, a subset of attributes, attributes that satisfy certain expression or value. Parameterizing both operators provides the option for (un)checking filtering of missing values.
5. **Data mining.** From the remaining table one can decide which attributes will be inspected for outliers. There are ways to find the outliers/errors in the table:
   (i) By inspecting the value of the data quality attribute which indicates the type of error in order to determine which contacts have the error in the attribute of interest or by filtering in/out rows with attribute fields containing out-of-range values.
   (ii) By using one of the outlier detection algorithms available in RapidMiner. RapidMiner supports density based cluster analysis, distance based cluster analysis, local outlier factors, and support vector machines. These can be found in the *Data Cleansing* group of operators. During the production of this report, an anomaly detection extension for RapidMiner was implemented and added to RapidMiner that contains the best known unsupervised anomaly detection algorithms. The *Anomaly detection* extension contains two categories of approaches: nearest-neighbor based and clustering based algorithms. Algorithms in the first category assume that outliers lie in sparse neighborhoods and that they are distant from their nearest neighbors. The second category operates on the output of clustering algorithms thus being much faster in general. Within the same extension, there are also statistical based (Histogram-based Outlier

Score (HBOS)) and performance score based (Receiver Operator Charateristic (ROC)) outlier detection algorithms.

The choice of algorithms depends on the attribute(s) that are inspected for outliers. Detecting outliers in RapidMiner is part of data pre-processing. If searching for contact outliers in position one can use distance based outlier detection algorithm. The process creates a new attribute **outlier**.

The fields in question can be treated as missing value fields. Their content can be replaced by average, minimum, maximum values, a specified value, zero or none. This can be done using *Replace Missing Values* operator.

The design processes in RapidMiner are described in Sections 6.2.1 and 6.2.2 while their XML codes can be found in the Appendix A.

6.2.1 Process Using Filter Example

Filter Examples operator can be used in a process for detecting invalid observations by checking the data quality or by directly inspecting the values of the attributes of interst. This can be done by parametrizing this operator with a string on the attribute value. The string could be for example, **data_quality** = 2, which would indicate out-of-range error for position either in longitude or latitude, or in both. The filter can also be created so as to

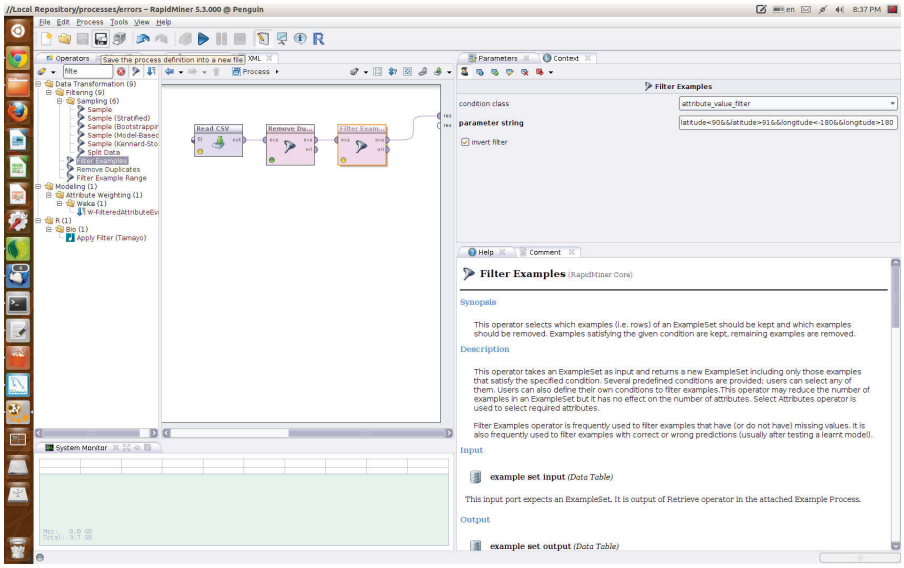

Figure 9: *Process which find out-of-range errors in (lon,lat) position by inspecting* **data_quality** *attribute.*

directly filter in/out the fields with out-of-range values for a single or multiple attributes, e.g. longitude, latitude, speed, etc. The parameter **condition class** should be set to attribute value filter, while the parameter string should contain the expression with attribute names and their out-of-range values. The process looks the same as in Figure 10, except for

the different parameter string, which here uses longitude and latitude attributes. Once the

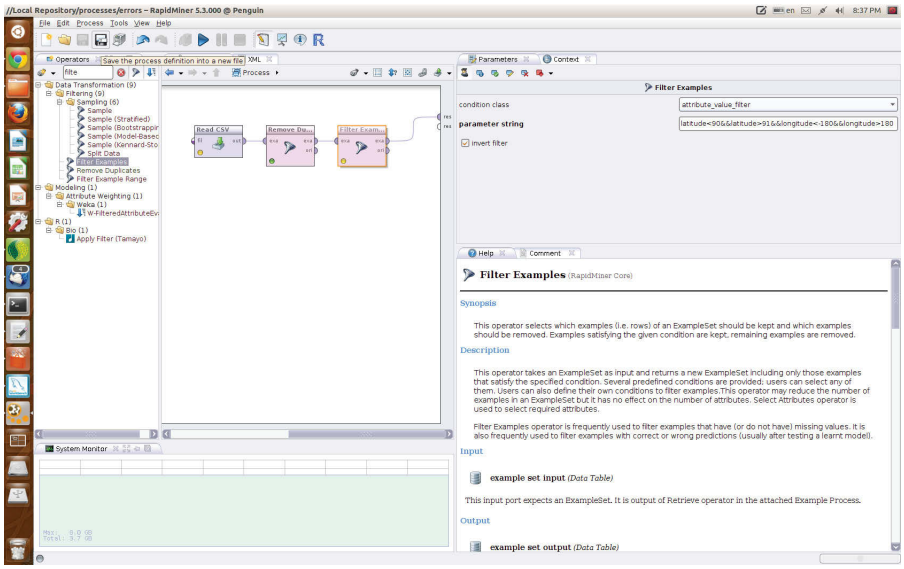

Figure 10: *Process to detect out-of-range errors in (lon,lat) positiion.*

process is run, the Example Set of results is created. The created dataset can be visualized by choosing Plot View button in the Example Set window, where one can choose Plotter, axes, color column, and other plot parameters. Figure 11 presents a visualization of data attributes longitude and latitude as read from **msari_error_total.csv**, colored by different MMSI.

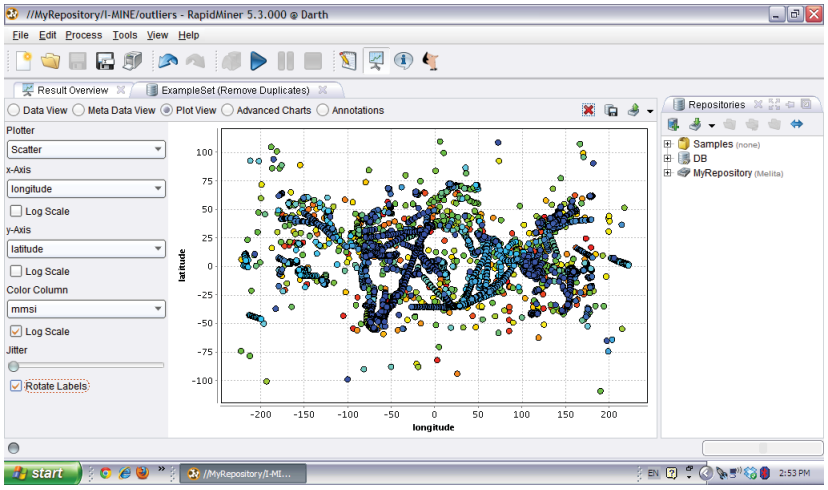

Figure 11: *Visualization of ExampleSet produced after removing the out-of-range errors and duplicates.*

6.2.2 Distance-Based Outlier Detection Process

The distance based (DB) outlier detection algorithm in RapidMiner calculates the DB(p,D)-outliers for an example set passed to the algorithm, as described by Knorr and Ng in [45]. A DB(p,D)-outlier is an object to which at least a proportion p of all objects are farther away than distance D. It essentially implements a global homogeneous outlier search.

6.3 Mining Ship Trajectories

Here, the design of several processes of mining ship trajectories using RapidMiner is described.

As in Section 6.2, before starting the design process in RapidMiner, we define the data mining steps as:

1. **Setting the target:** The process is to mine the following pattern:
   A ship is spotted in port X and may be transiting to port Y for time period T. Identify all ships that may fit that description.
   under the assumption that if a ship is observed in port A at earlier time than it is observed in port B then there is a route between port A and port B.
2. **Establishing the target dataset.** For this data mining task, the data set has been provided provided in the file **contacts.csv**, which contains AIS data with the following attributes:

contact ID	MMSI	longitude	latitude	report_timestamp	data_quality

   The file **contacts.csv** has been created from MSARI DB as described in Section 5.
3. **Data preprocessing.** To facilitate processing in RapidMiner the grid has been created, which covers the globe and maps (lon,lat) pairs to $1^0 \times 1^0$ cells, each labeled with grid cell ID. The data have been manipulated so as to contain the cell_ID attribute corresponding to each (lon,lat) pair.

MMSI	longitude	latitude	report_timestamp	cell_ID

   Next, file **port_to_cell.csv** which contains mappings of relevant ports and their neighborhoods to cells has been created.

port	latitude	longitude	cell_ID

   Removing the out-of-range errors and rows with other invalid and missing values make part of this step.
   Finally, a local database **Contact_info**, containing all the data with all attributes, has been created to speed up the processing with RapidMiner.

Row.No	MMSI	latitude	longitude	cell_ID	port	report_time

4. **Data cleaning.** Adding and removing unwanted attributes has already been done within the previous step. Removing contact duplicates within the same cell is part of this step.

5. **Data mining.** Two approaches have been attempted:

   (i) Association rule mining. Every route background is different and every grid's cell characteristics is different. The data mining approach here is to discover route behaviors by discovering the high traffic density grid cells. This is similar to to discovering consumers' behavior and discovering popular frequent products in order to create a marketing strategy. Each customer' s basic information is equivalent to the static message, the transaction data to route related message, ship's MMSI to customer's number, every product list in a transaction equivalent to grid cells passed along the route. The most popular approach to discovering hidden patterns in these kind of situations is association rule mining. Having AIS data and route related messages, we can assume that $I = \{i \subset 1, i \subset 2 \cdots i \subset m\}$ represent all grid cells in the area, called *itemsets*, and every route R is a subset of all itemsets, i.e. $R \subset I$. The association rule *Ship with some kind of profile who normally passes region A will also pass region B* can then be formally written as follows:

$$\begin{aligned} A \rightarrow B[support, confidence] & \\ support &= P(AB) \\ confidence &= P(AB)/P(B) \\ A \subset I, B \subset I and AB &= 0 \end{aligned}$$

   (ii) Manipulating data, using SQL queries, and creating new data.

   Design processes, vizualization and analyses of results, as final steps of data mining, are described in Sections 6.3.1 and 6.3.2.

6.3.1 Mining Association Rules Process

Mining regularities of AIS data was attempted in RapidMiner using *FP Growth* operator which calculates itemsets from the given ExampleSet using FP-tree data structure by Han et al. (2000). It is compulsory that all attributes of the input data to this operator are binominal. The itemsets corresponding to each MMSI are subsequently fed into *Create Association Rules* operator to produce rules. The frequent itemsets for each MMSI is created using *Loop Value* operator which provides an iteration macro that stores current MMSI value. The value is accessible inside the loop. In the loop, to obtain the MMSI, *Filter Example* operator has to be parametrized so that the parameter string is $mmsi = \%(mmsi\_id)$

Even with lowering confidence to 0.0001, no rules were created. The process design flows are shown in Figures 12 and 13 respectively.

Another attempt to discover association rules was made with WEKA's *W-Apriori* operator, which implements an Apriori-type algorithm which iteratively reduces the minimum support until it finds number of rules with the given minimum confidence. This is the original

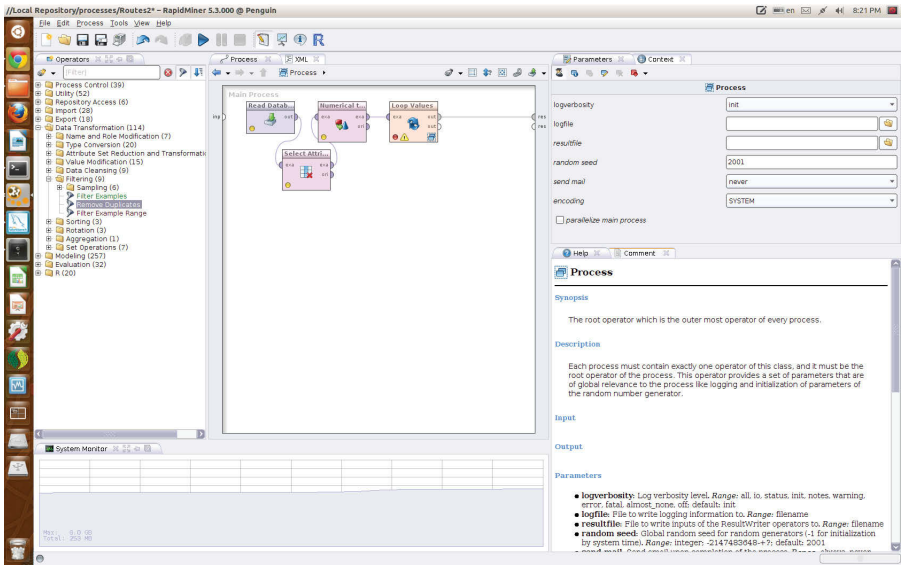

Figure 12: *Association rule process.*

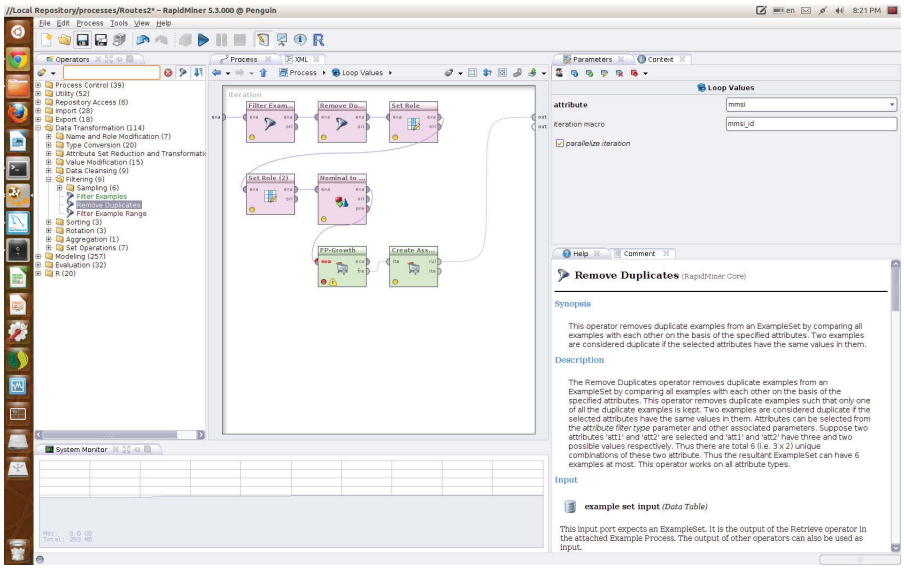

Figure 13: *Creating frequent itemsets with FP-Growth and creating association rules within MMSI loop.*

Agrawal's algorithm[9] for market basket analysis. This algorithm did not produce any rules either, which may be due to very small sizes of the itemsets.

Since the itemsets contain contacts ordered in time, it seemed appropriate to attempt sequential pattern mining. Sequential pattern mining is similar to association rule mining with the difference in consideration of the time relationship of data i.e. the order in time in which events happen. For example, if $A \rightarrow B$ then A must happen before B, which is a

casual relationship, while association rule mining is concerned with which events happen at the same time. RapidMiner supports WEKA's *W-GeneralizedSequentialPatterns* operator/algorithm, which permits specifying the attribute(s) that have to be contained in each itemset of a sequence. This process has large memory requirements (as seen in System Monitor) and was not able to be completed in reasonable amount of time due to limited resources.

6.3.2 SQL Query Based Process

Given the format of our initial data, RapidMiner did not lend itself well to route identification. We therefore proceeded to transform the data further using the SQL query language and creating additional data.

After removing duplicates from all cells with *Remove Duplicates* operator, the SQL query was created to identify routes from **Contact_info** local DB.

The query produces the Example Set shown in Table 5

MMSI	port_a	port_b	time_a	time_b
209656000	MONTREAL	QUEBEC	11/04/11 04:29 AM	11/07/11 05:15 AM
209997000	MONTREAL	SIDNEY	11/07/11 03:44 AM	11/10/11 04:33 AM
211727000	NEW YORK	HAMPTON	11/05/2011 03:46 PM	11/08/11 03:44 PM
⋮	⋮	⋮	…	…

Table 5: *Identified ship routes from port_a to port_b*

To select routes which correspond to certain periods of time, this table may be exported to a CSV file, then, one can create a process with *Read CSV* and Filter Examples or *Filter Example Range* operators. The *Filter Example* operator must be parameterized to contain a string which defines the desired beginning and end of the period of interest using the attribute *report_timestamp*. The same applies to *Filter Example Range* which selects which rows within the specified index range should be kept/removed. Note that if date_time format is chosen for date and time, parsing must be carefully done. For changing the granularity of time stamps to day or a week, or month by using the operator *Date to Nominal* can be used. The other solution involves parsing numbers out of report_timestamp attribute by changing the type of the attribute to numeric.

6.4 RapidMiner Integration with a SQL Server and SQL Queries

Both open source and commercial versions of RapidMiner can be integrated with all types of SQL servers (Oracle, MySQL, Microsoft(MS) SQL). For MS SQL server running on the local machine, it is necessary to ensure that the browser services are running and that the latest jTDS-SQL Server and Sybase JDBC driver [32] is installed.

For any SQL server, a DB connection must be created in RapidMiner by clicking on Tools → Manage Database Connections. Once the DB connection is created, its content can be read using *Read Database* operator in RapidMiner. The operator's parameter **connection** should be set to the corresponding DB, while **define query** in this case should be set to table name, while **table name** should be set to **Contact_info**. The SQL queries can be

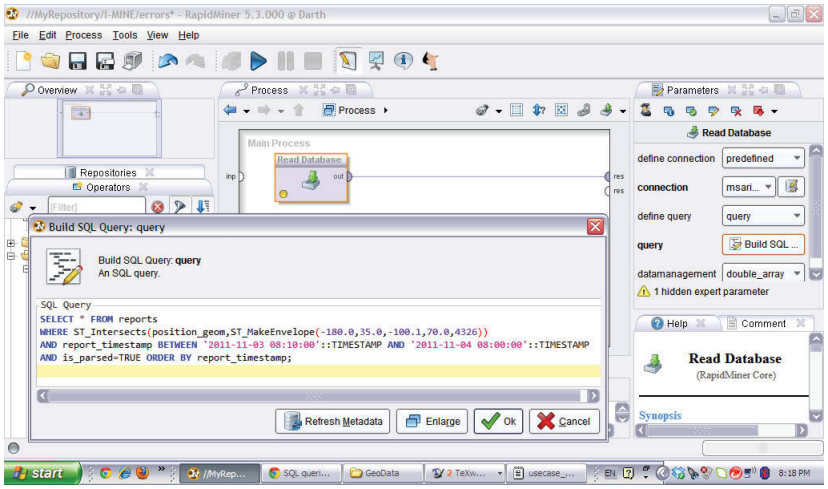

Figure 14: *Connecting to a DB and creating SQL queries.*

built in the same operator parameters' window by clicking on Build SQL query or defined from a file or a table. Once the process is run, the results can be viewed in the Results View of RapidMiner. There is also a URL option for connection, which requires entering username and password. Another way to connect is through File → Import Data menu, where one can select the pre-existing connection to connect, define a query and store data in RapidMiner's local repository.

SQL statements can also be executed through Execute SQL operator which performs an arbitrary SQL statement on an SQL database (see Fig. 15). The SQL query can be passed to RapidMiner via a parameter or, in case of long SQL statements, in a separate file. This operator cannot be used to load data from databases but merely to execute SQL statement. In order to load data from a database, the operators *Read Database*, *DatabaseExampleSource* or *CachedDatabaseExampleSource* can be used.

32. found at `http://www.sourceforge.net`

RapidMiner does not allow data manipulation statements with *Execute Query* operator. All data manipulation must be done prior to this operator.

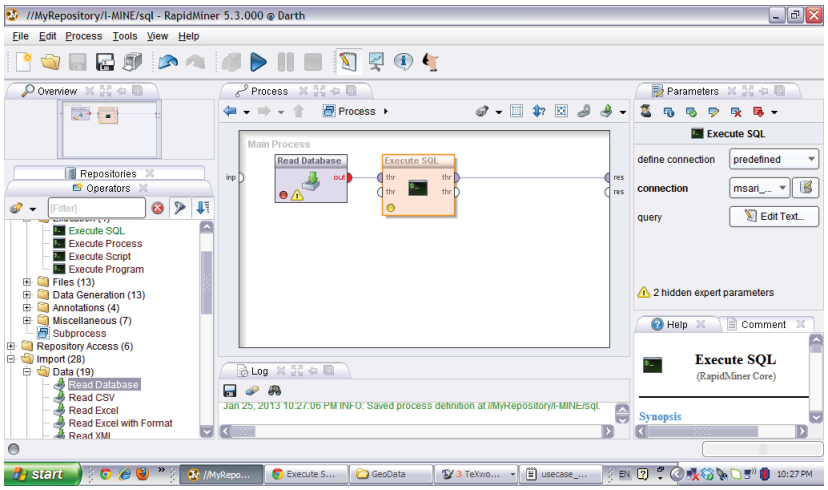

Figure 15: *Process with an SQL statement using* Execute SQL *operator.*

6.5 General Remarks

RapidMiner consumes a vast amount of system resources. Namely, it generally consumes a large span of available RAM for mining operations. Although RapidMiner claims that operators such as the ”Stream DB” allow it to connect in a cached data mode thereby reducing its memory footprint, our trials have shown this operator behaves in an unstable fashion. It is advisable that users of RapidMiner who intend to work with large datasets do so on systems with large amounts of RAM (above 4GB) and that they set the appropriate Java Virtual Machine(JVM) parameters (MAX_JAVA_MEMORY) to allow the JVM to use more RAM than standard.

Most of the learning schemes and algorithms in RapidMiner require the attributes or labels to be of a certain type, therefore, it is necessary to transform the data accordingly, which may be challenging at times. Also, parameterizing different operators may use different syntax. In general, even with all GUI functionalities, the data preparation process is time consuming.

Memory problems encountered with RapidMiner make it difficult to deal with large sets of data and successfully mine for knowledge, making large memory resources essential.

Although at first sight RapidMiner appears easy to use because of its functional GUI (process flow elements can be easily added, exchanged or disabled in a particular process run), and extensibility (there is a number of available learning algorithms, both RapidMiner’s and third party) the interface learning curve is rather steep. **It is very useful and highly**

recommended to be familiar with the data, data structures, and machine learning schemes and algorithms to be able to choose the right operator/algorithm, hence optimally design a data mining process.

As for spatio-temporal data processing, RapidMiner is found to be suitable for data manipulation and visualization only. This is in agreement with previous uses of RapidMiner where in-house developed algorithms have been combined with RapidMiner visualization capabilities, such as in [46].

7 Maritime Traffic Data Mining with R

R is an programming language supporting procedural, object-oriented, array and functional programming paradigms. It is typically used through a command line interpreter but also through scripting. An R script is simply a text file containing the commands to be executed on the command line allowing users to automate procedures.

The R environment comes with the interpreter and a basic script editor. More sophisticated editors and IDE are available with support for R such as the Eclipse plugin StatET [33].

Several GUIs are also available to interface with R, some designed for specific tasks such as Rattle for data mining. See [47] for a list of all available GUIs for R. In this investigation, only the native command line interpreter and Rattle were used.

The main advantages of R for data mining are:

1. Ease of data manipulation: R supports array programming making it very efficient for array, matrix and other R array-based objects manipulations (by avoiding loops).
2. Extensibility: The capabilities of R are extended through user-created packages, which allow specialized statistical techniques, visualizations, import/export capabilities, reporting tools, bridges to other languages (Java, Python, C,...), etc.
3. Very active community of developers, including expert statisticians: there are many resources freely available and of high quality and packages are frequently updated.

The learning curve may be steep, but the massive amount of references, forums and mailing lists eases the learning process. Note that a scientist familiar with Matlab will feel comfortable with R, mainly because of the syntax and programming paradigm. On that topic, see [48] for a comprehensive comparison between Matlab and R functions.

7.1 Data Mining Work Flow with R

Typically, the work flow of data mining with R is:

1. Import the required packages
2. Import data
3. Transform data in a convenient format for analysis
4. Use the data mining/statistical functions and/or Rattle.
5. Visualize, validate and export results.

The following paragraphs describe each of these steps.

33. `http://www.walware.de/goto/statet`

7.1.1 Step 1: Package Import

As mentioned above, capabilities of R are extended through packages. These packages are available from the Comprehensive R Archive Network (CRAN). CRAN is a network of File Transfer Protocol (FTP) and web servers around the world that store identical, up-to-date, versions of code and documentation for R. CRAN is accessible from the R environment, allowing packages installation and load with only two commands:

```
> install.packages("name_of_library")
> library(name_of_library)
```

7.1.2 Step 2: Data Import

There exist several packages allowing users to load different kinds of file formats and interact with databases. In this investigation, data were loaded from CSV files and directly from MSARI which is managed by PostgreSQL.

Importing data from a CSV file (or similarly a text file) is done with a single command and no package is required:

```
> msari_data <- read.csv(file="C:\\Users\\...\\msari_error_total.csv",
                head=TRUE,sep=",")
```

Importing data directly from MSARI with R is simple and very efficient. The packages `DBI`[34] and `RPostgreSQL`[35] are required to interact with PostgreSQL. Data can also be imported from Microsoft SQL Server, using ODBC with RODBC[36] for Windows and JDBC with RJDBC[37] for Linux.

A connection to the database is opened with:

```
> drv <- dbDriver("PostgreSQL")
> con <- dbConnect(drv, dbname = "msari_db", host='192.000.00.000',
         user='postgres', password='dummy',port='5432')
```

The query is defined, sent and results are fetched with the following commands:

34. `http://cran.r-project.org/web/packages/DBI/index.html`
35. `http://cran.r-project.org/web/packages/RPostgreSQL/index.html`
36. http://cran.r-project.org/web/packages/RODBC/index.html
37. http://www.rforge.net/RJDBC/

```
> sqlQuery = "SELECT * FROM reports WHERE mmsi = 636091992"
> rs <- dbSendQuery(con, statement = paste(sqlQuery))
> msari_data <- fetch(rs, n = -1)
```

It is also possible to connect to Oracle, MS SQL Server and MySQL in similar ways, using the appropriate package.

During this investigation, no R package offering an SQL query builder, at the command line or with a GUI, was found. It means that with R, all queries to a RDBMS have to be explicitly written in SQL.

To conclude on data import, let us mention that PL/R is a promising option. It is a procedural language for PostgreSQL that allows to write PostgreSQL functions and triggers in the R programming language (instead of using the PL/pgSQL scripting language) and it offers most of the capabilities of the R language. It means that for a database such as MSARI, users could have access, with an SQL query, to statistics functions for Maritime Situational Awareness (MSA) data analysis.

7.1.3 Step 3: Data Pre-Processing and Transformation

As mentioned in the introduction, data preparation is one of the most time consuming steps of the KDD. R supports array programming. This step is greatly simplified since R is optimized for operations on array-based objects. It means no looping on array dimensions is required, speeding up and simplifying all array-based data objects manipulations.

Note that if data are originally stored in a relational database, then part of data manipulation can be carried out with SQL. A scientist who is used to SQL may feel more comfortable executing some data manipulation with SQL than with R.

7.1.4 Step 4: Data Mining

Through its packages, R offers a wide range of DM capabilities and Rattle is a good start for an initiation to data mining with R. It allows users to explore some data mining capabilities available in R, without the burden of looking for available packages. However Rattle exposes only a subset of all R DM capabilities. So the data miner will most probably feel limited by the GUI functionalities and will want direct access to the functions. This was the case for both scenarios investigated. Moreover, Rattle does not allow users to manipulate the output of the DM analysis, other than through its offered functionalities. For instance, it is not possible to transform the output of a clustering operation to use it as input for another algorithm. This limitation may be critical for a custom data mining process.

7.1.5 Step 5: Results Visualization, Validation and Export

Some visualization and export functionalities are available from Rattle. There are also many data visualization packages available. Because most data mining functions produce a data set as output, this data set can be used as import (sometimes with some manipulation required) for visualization or other analysis functions.

Note that R offers no means to validate the results. It is the data miner's responsibility to develop methods to make sure that the results are realistic. That can be done using visualization, statistical tests or external validated results. This aspect of data mining is not specific to R. Many DM algorithms produce results that require an understanding of DM concepts or at least some basic statistics knowledge.

With R, it is possible to export data in almost any format: CSV, Excel, text, XML, etc. It is also possible to store back the results in a database, using the appropriate package - e.g. `RPostgreSQL`.

7.2 Mining Invalid Observations

This Section describes the data mining process developed with R to answer the following question: *If an out-of-range error is identified in one field, is it possible to determine if other fields are likely to also have an out-of-range problem?* For instance, when a vessel heading is identified as invalid, is there a greater likelihood that other fields such as latitude or speed, are also invalid?

This analysis was performed on the AIS data set first described in Section 5.

The AIS specifications (see [49]) describe for each field the range of possible values. For instance, heading (available from messages of type 1, 2, 3, 18 and 19) takes values between 0 and 359 and has the value 511 when is not available. Therefore, for AIS data, out-of-range values can be determined explicitly, by verifying if the value of the field is in the specified range.

The basic idea was to explore the co-frequency of invalid values occurrence. For each pair of fields, the frequency at which a field A has an invalid value when field B has an invalid value, $\mathcal{F}(A|B)$, is computed. It is important to note that the measure estimated is the *frequency* and not the *probability*. Estimating the conditional probability would require estimating the distribution of invalid values for AIS, which is out of the scope of this investigation.

Computing $\mathcal{F}(A|B)$ is straightforward. It is the conditional sum of all out-of-range values for A across all reports in the data set normalized by the number of invalid values for field

B:

$$\mathcal{F}(A|B) = \frac{\sum_{i=1}^{N} \delta(A_i).\delta(B_i)}{\sum_{i=1}^{N} \delta(B_i)}, \tag{1}$$

where N is the number of reports in the data set and

$$\delta(A_i) = \begin{cases} 1, & \text{report } i \text{ as an invalid value at field A,} \\ 0, & \text{otherwise.} \end{cases} \tag{2}$$

7.2.1 Work Flow

The analysis is performed following the work flow described in Section 7.1. For this analysis, no extra package is required and data is imported from the MSARI DB as described in Section 7.1.2.

The output of the data import is a data frame containing a list of reports with MMSI, time stamps and attributes mentioned above. Some values are empty; for instance, there is no IMO number for AIS message types 1, 2 and 3. A data frame is an R data object, which is a type of table where the typical use employs the rows as observations and the columns as variables.

The data has to be transformed to get a data frame populated of $\delta(X_i)$ (as in equation 2), i.e. to replace by 1 each out-of-range or invalid value and by 0 each valid value. This is done by encoding the out-of-range and invalid values specifications in R.

The data mining step is performed by computing $\mathcal{F}(A|B)$ as described by equation 1. This may not be considered *data mining* as described in Section 2 of this document. However, no sophisticated algorithm was required to get the information needed. Most of the efforts are in the data import/manipulation steps.

All these operations were performed without any memory issue and all computations were very fast.

The R commands are listed in annex C.1.

7.2.2 Results

The results of this analysis are presented in Table 6. Only fields where invalid values were found in the data set are listed in the table. Each cell (except the diagonal ones) represents a value of $\mathcal{F}(A|B)$. It reads from rows to column -e.g. $\mathcal{F}(Heading|Longitude) = 96.39\%$. In other words, based on AIS data reported from exactEarth from November 3 to November 10 2011, when an invalid longitude value is observed, 96.39% of the time the heading is also invalid.

A \ B	Longitude	Heading	Course	Ship Type	MMSI	IMO	ETA Month	ETA Hour
Longitude	-	2.46%	7.94%	0%	0.36%	0%	0%	0%
Heading	96.39%	-	23.81%	0%	65.14%	0%	0%	0%
Course	0.56%	0.04%	-	0%	0.29%	0%	0%	0%
Ship Type	0%	0%	0%	-	2.00%	85.71%	3.28%	1.79%
MMSI	0.56%	2.62%	6.35%	14.14%	-	0%	0%	0%
IMO	0%	0%	0%	45.45%	2.00%	-	24.59%	25.00%
ETA Month	0%	0%	0%	1.01%	0%	0%	-	87.50%
ETA Hour	0%	0%	0%	0.51%	0%	0%	80.33%	-

Table 6: *Values of $\mathcal{F}(A|B)$ for the AIS data set reported by exactEarth and decoded and parsed by MSARI. It reads from row to column, e.g. $\mathcal{F}(Heading|MMSI) = 65.143\%$*

Table 7 contains the total count and proportion of invalid values in the complete data set for each field.

Field	Proportion	Count
Longitude	0.0205%	887
Heading	0.8058%	34,825
Course	0.0015%	63
Ship Type	0.0046%	198
MMSI	0.0324%	1,400
IMO	0.0024%	105
ETA Month	0.0014%	61
ETA Hour	0.0013%	56

Table 7: *Total count and proportion of invalid values in the complete data set for each field were invalid values were found.*

7.2.3 Alternative Methodology

In the case where out-of-range values are unknown, e.g. for a new data type with unknown specifications, outliers detection approaches could be used. Based on the assumption that invalid values are outliers, i.e. observations significantly distant from other data, these methods would detect automatically the out-of-range values for each fields. Once these values are identified, the same methodology as above could be reused. There are several R packages for outliers detection. The following have been explored for this investigation and found convenient and robust to the data set used: `outliers` [38] and `extRemes` [39] which has a GUI.

38. `http://cran.r-project.org/web/packages/outliers/index.html`
39. `http://cran.r-project.org/web/packages/extRemes/index.html`

7.3 Mining Ship Trajectories

This Section describes the data mining process developed with R to analysis some aspects of the spatio-temporal aspect of AIS data. It explores ways to extract ship movements between ports by answering the following questions:

1. *Identify all ships that transited from port X to port Y, for a time period T.*
2. *If a ship is spotted in port X, in what port it is the most likely to be transiting to?*

The R work flow for the analysis of question 1 is referred as sub-case 1 and the one for question 2 is referred as sub-case 2 and are described in Sections 7.3.1.1 and 7.3.1.2 respectively.

The analysis was performed using the `contact_to_cell` and `port_to_cell` files as data sets described in Section 5.

Note that it is possible to build and manipulate a geographical grid with R, with package `gdistance` [40]. For this analysis, it was however decided to created that grid and map positions to cells using PostGIS.

7.3.1 Work Flow

The analysis of ship movement between ports is also performed following the work flow described in Section 7.1.

The data is imported from a CSV file, as described in Section 7.1.2. Note that the data could have been imported directly from MSARI DB as for the first scenario.

The output of the reports data import is a data frame containing a list of reports with MMSI, time stamp, latitude, longitude and cell ID.

Only MMSI and cell IDs are required for the analysis, the others fields are removed from the data frame. Also, all duplicated entries of the resulting data frame have to be removed to perform the analysis. A duplicate is defined by a ship reported in the same cell for more than one consecutive time. Only the unique cells visited by each ship are required for the analysis. R commands used to transform the data are listed in annex C.2.

At this point, the data frame contains a list of MMSI with associated cell IDs, ordered in time, without duplicates, as illustrated in Table 8.

7.3.1.1 Sub-Case 1

This analysis aims at identifying all ships that transited from port X to port Y, for a time period T. With R, this analysis is very simple to perform. Ship that passed by the cell

40. `http://cran.r-project.org/web/packages/gdistance/index.html`

MMSI	cell id
211205790	66451
210938000	67799
210938000	68243
210938000	67798
210938000	66907
212236000	66008
212236000	66009

Table 8: *Excerpt of the data frame used for the port transiting analysis.*

associated to port X and then to port Y, during time period T, satisfy this description. To simplify the notation, consider cell X as the cell corresponding to the position of port X and and cell Y as the one corresponding to port Y.

Let us suppose that at this point the data frame contains only contacts from the time period T and is ordered by MMSI and time (as illustrated in Table 8). That filtering on time and ordering can be done on MSARI with SQL and also with R using the Date class. The R Date class allows to transform a string to a Date object. From there, filtering on date is straightforward:

```
> mydata_filtered
   <- mydata[mydata$mydate %in% as.Date(c('2012-01-05', '2012-01-09')),]
```

At this point, it is a matter of filtering data to get only MMSIs associated to cells X and Y and make sure that it is ordered properly, i.e that port X was visited before Y. This is achieved with two R commands listed in Annex C.2.1. This manipulation provides the list of MMSI that has transited to port X and then Y. One could have also added the extra restriction that the ship must not visit other ports between X and Y.

For instance, based on the AIS data set used, between November 3 and 10 2011, there is a total of 26 ships that visited the port of Montreal and then Quebec; and 22 the other way.

7.3.1.2 Sub-Case 2

This analysis aims at identifying the most probable links between ports, based again on the AIS data set. Knowing the most probable links between ports enable answering questions such as: *If a ship is spotted in port X, in what port it is the most likely to be transiting to?*

That kind of analysis is typical in marketing and is often referred as the market basket analysis. It is a modelling approach based on the assumption that if someone buys a certain group of items, that someone is more likely to buy another group of items, e.g. chips and cola. This assumption can be transposed to the port case: if a ship transits by a given port,

it is most likely to transit to an other given port, e.g. Montreal and Quebec. The market basket analysis, also called the association rule mining, seeks to find relationships between items. It outputs a list of association rules providing information in the form of *if-then* statements, such as *if* port X is visited, *then* port Y will be visited.

This analysis was performed using the `arules` [41] and `Rattle` package and GUI. Note that the basket analysis offered by rattle is performed by the `arules` functions. Therefore, using Rattle or `arules` functions will provide the same results. However, it was found that carrying out the analysis with Rattle has limitations. For instance, it is not possible to specify the length of the rules and it is not possible to manipulate the rules once computed. For this analysis, it was required to filter the rules resulting from the analysis, so the final work flow was performed without Rattle.

The first step of the data manipulation process is to remove all ships with only one contact. Thus, only ships with 2 contacts or more are considered for this analysis. There are 1,029 such ships in the data set. Moreover, the use of `arules` association algorithm requires some non trivial preliminary data manipulations. The algorithm takes as input an incidence (or frequency) matrix. In the current case, it is a matrix with MMSI for rows and cell ID as column. For instance, there would be a 1 in the matrix cell (967191190, 65119) if the ship with MMSI 967191190 has a contact within the cell of ID 65119, and 0 otherwise. Since all duplicates have been removed, the matrix is only populated by 0 and 1.

The association algorithm, called `apriori`, produces a set of rules, based on the user provided support, confidence, minimum and maximum length of the rules. The smaller support and confidence are, the higher the number of rules produced will be.

With a confidence of 0.1 and a support of 0.01 and rules of maximum length 2, 827 rules are produced. From this set of rules, only rules that imply the list of ports of interest need to be considered.

Results are summarized in Table 9, ordered by lift and confidence. The first rule means that if a ship is spotted at the Baltimore port and is transiting to another port, then this port is most likely to Hampton.

The `arulesViz` package [42] allows users to visualize rules with different methods: matrix, graph, etc. Figure 16 is a graph representation of the rules listed in Table 9, where Hampton corresponds to cell ID 419, Baltimore to 324, Montreal to 1597, Quebec to 2538, New Haven to 1788, New York to 1370, Philadelphia to 847 and Marcus Hook to cell ID 742.

Note that if we increase the minimum number of contacts for each ship, rules change. For instance, if we consider only ships with at least 20 contacts, which decrease the number of ships for the analysis, the rules implying ports become:

41. `http://cran.r-project.org/web/packages/arules/index.html`
42. `http://cran.r-project.org/web/packages/arulesViz/index.html`

If	Then	Lift	Confidence	Support
Philadelphia (PA)	Marcus Hook (PA)	9.615	0.585	0.03109
Marcus Hook (PA)	Philadelphia (PA)	9.615	0.511	0.03109
Quebec (QC)	Montreal (QC)	5.147	0.333	0.0117
Montreal (QC)	Quebec (QC)	5.147	0.180	0.0117
New Haven (CT)	New York (NY)	4.536	0.652	0.0194
New York (NY)	New Haven (CT)	4.536	0.135	0.0194
Baltimore (MD)	Hampton (VA)	2.853	0.243	0.0129
Hampton (VA)	Baltimore (MD)	2.853	0.152	0.0129
Philadelphia (PA)	New York (NY)	1.866	0.268	0.0142

Table 9: *Rules, implying ports, as computed by the association algorithm.*

– Quebec ⇒ Montreal, confidence = 0.353;
– Montreal ⇒ Quebec, confidence = 0.300;
with lift 2.7 and support of 0.0392.

To conclude this scenario analysis, it is worth mentioning that results may have been different with a finer grid. Also, for a finer analysis, one could have filtered ships by types - e.g. to get only cargo.

Finally, `arules`, basic R functionalities used with PostGis were sufficient to explore that scenario. The next step would be to explore capabilities offered by the spatio-temporal packages listed in Section 4.1.5.2. These packages typically deal with the extra complexity of the multi-dimensional aspects of this type of data, such as spatio-temporal auto correlation (see Section 2.2). However, the statistical tools offered in such packages require a good understanding of statistics and spatio-temporal statistics.

All R commands used for that sub-case are in the Annex C.2.2.

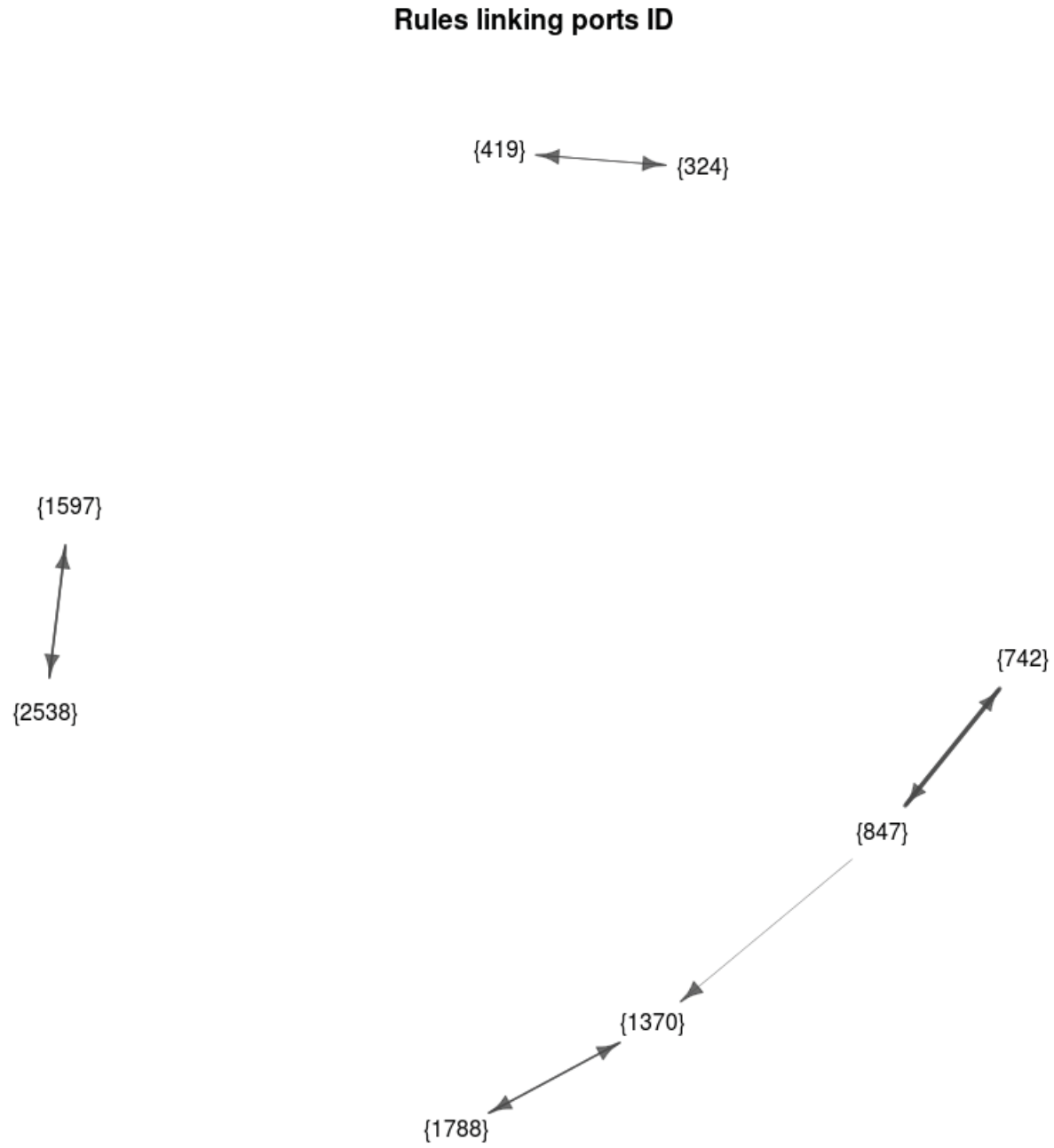

Figure 16: *Visualization of the rules described in Table 9 as a network of ports.*

8 Assessment Summary of RapidMiner and R for Maritime Traffic Data Mining

This Section concludes with a summary of the findings about RapidMiner and R and some concluding remarks about a possible integration within an operational context, such as at MARLANT.

Table 10 summarize findings on RapidMiner, detailed in Section 6 and findings on R, detailed in Section 7.

Table 10: Summarizing comments about RapidMiner and R for AIS data mining.

	RapidMiner	**R**
Ease of Use	GUI easy to use, but optimal process design requires familiarity with the data, data structures, and machine learning schemes and algorithms	Steep learning curve, because it requires users to learn the language to fully benefit from R
Database Connectivity	Allows user to connect and retrieve data from all major Database Management System (DBMS)	Allow user to connect and retrieve data sets from major DBMS
Scalability	Working with large datasets requires systems with large amounts of RAM (above 4GB); it is highly recommended to set the appropriate Java Virtual Machine(JVM) parameters (MAX_JAVA_MEMORY) to allow the JVM to use more RAM than standard.	Handles large data sets and extensive queries
Speed	Complex and nested processes require longer processing time	Setup and execution are fast
Extensibility	A number of third party packages are available, including R extension	Capabilities extended through packages

Continued on next page

Table 10 – continued from previous page

	RapidMiner	R
Results validation	Depends on the amount of data to be processed and user's knowledge about target data set and the algorithms used in the design process.	Depends on the packages used, but validation is the user's responsibility
Visualization and export	Easy visualization and export.	Many packages for visualization and reporting.
General DM Capabilities	Generally good for other than spatio-temporal data	Very good
Spatio-Temporal DM Capabilities	Poor except for data preprocessing and cleaning. Good for visualization.	Partially explored using grid strategies - some spatio-temporal packages look very promising
SQL GUI Development	It can be integrated with a SQL server. SQL statements can also be executed through Execute SQL operator which performs an arbitrary SQL statement on an SQL database	None. All interactions with an RDBMS have to be in explicitly written in SQL

The most challenging issues in maritime traffic data mining prove to be the volume of data and their spatio-temporal characteristics. These should be the decision making factors in choosing (or developing) the appropriate data mining tool for this kind of data.

The results of this preliminary evaluation of RapidMiner and R in application to AIS data mining give preference to R over RapidMiner as it has less memory requirements and provides easy and important functionality for manipulating all, and especially temporal, aspects of AIS data.

A more detailed investigation using other specific maritime traffic scenarios of both selected data mining tools is recommended to further confirm these preliminary findings. Also, developing a custom application with a GUI (e.g. in Java) combined with R is a promising option as such tool would compensate for the GUI limitations of R/Rattle.

References

[1] Fayyad, U., Piatetsky-Shapiro, G., and Smyth, P. (1996), From Data Mining to Discovery Knowledge in Databases, *AI Magazine*, 3(17).

[2] Ristić, B., La Scala, B., Morelande, M., and Gordon, N. (2008), Statistical analysis of motion patterns in AIS Data: Anomaly detection and motion prediction, Proceedings of the 11th International Conference on Informatin Fusion.

[3] Hammond, T., McIntyre, M., Chapman, D., and Lapinski, L. (2006), The Impact of Self-Reporting Systems on Maritime Domain Awareness, Proceedings of the 11th International Command and Control Research and Technology Symposium.

[4] Coenen, F. (2011), Data mining: past, present and future, *The Knowledge Engineering Review*, 26(1), 25–29.

[5] M.-O., St-Hilaire, M., Mayrand, and D., Radulescu (2012), Maritime Situational Awareness Research Infrastructure (MSARI): Requirements and High Level Design, Technical Report DRDC Atlantic.

[6] Frawley, W., Piatesky-Shapiro, G., and Matheus, C. (1991), Knowledge discovery in database: an overview, Fayyad, U.M. and Piatesky-Shapiro, G., Smyth, P. and Ulthurusamy, R. (Eds.), Knowledge Discovery in Database, MIT Press, Cambridge, MA.

[7] Li, T., Ding, C., and Wang, F. (2011), Guest editorial: special issue on data mining with matrices, graphs and tensors, *Data Mining and Knowledge Discovery*, 22, 337–339.

[8] Han, J. and Kamber, M. (2003), Data Mining: Concepts and Techniques, Morgan Kaufman, San Francisco.

[9] Agrawal, R., Imielinski, T., and Swami, A. (1993), Mining association rules between sets of items in large databases, ACM SIGMOD International Conference on Management of Data, pp. 207–216.

[10] Han, J., Pei, J., and Yin, Y. (2000), Mining frequent patterns without candidate generation, Proceedings of the ACM SIGMOD Conference on Management of Data, SIGMOD'00), pp. 1–12.

[11] MacQueen, J. B. (1967), Some Methods for Classification and Analysis of Multivariate Observations, Proceedings of 5th Berkeley Symposium on Mathematical Statistics and Probability, pp. 281–297.

[12] Hastie, T. and Tibishrani, R. (1996), Discriminant Adaptive Nearest Neighbor Classification, *IEEE Transactions on Pattern Analysis and Knowledge Intelligence*, 18(6), 607–616.

[13] Zhang, T., Ramakrishnan, R., and Livny, M. (1996), BIRCH: An Efficient Data Clustering Method for Very Large Databases.

[14] Fisher, D. (1987), Improving inference through conceptual clustering, Proceedings of the 1987 AAAI Conference, pp. 461–465.

[15] Rastogi, R. and Shim, K. (1998), A decision tree classifier that integrates building and pruning, Proceedings of the Int'l Conference on Very Large Databases.

[16] Mascaro, S., Korb, K., and Nicholson, A. (2010), Learning Abnormal Vessel Behaviour from AIS Data with Bayesian Networks at Two Time Scales (online), Germain Research Center for Artificial Intelligence, `http://www.bayesian-intelligence.com/publications/TR2010_4_AbnormalVesselBehaviour.pdf` (Access Date: 2012).

[17] Laxhammar, R. (2008), Anomaly detection for sea surveillance, Proceedings of the 11th Interna- tional Conference on Information Fusion, pp. 55–62.

[18] Li, X., Han, J., and Kim, S. (2006), Motion-Alert: Automatic anomaly detection in massive moving objects, Proceedings of the IEEE Intelligence and Security Informatics Conference (ISI 2006), pp. 166–177.

[19] Roiger, R. and Goetz, M. (2002), Data Mining: A Tutorial Based Primer, Adison Wesley.

[20] Shekhar, S., Evans, M.R., Kang, J.M., and Mohan, P. (2011), Identifying patterns in spatial information: a survey of methods, *John Wiley & Sons, Inc. WIREs Data Mining Knowledge Discovery*, 1, 193–214.

[21] Miller, H. J. (2008), Handbook of Geographic Information Science, Eds: J. P. Wilson and A. S. Fotheringham, Blackwell Publishing Geospatial Ontology Development and Semantic Analytics.

[22] W., Tobler (1970), A computer movie simulating urban growth in the Detroit region, *Economic Geography*, 46(2), 234–240.

[23] Reed, Greg (2006), Extending the Weka Data Mining Toolkit to support Geographic Data Preprocessing, (Technical Report RP-354) Instituto de Informatica - UFRGS, Porto Alegre.

[24] Shekhar, S., Vatsavai, R.R., and Celik, M. (2008), Spatial and Spatiotemporal Data Mining: Recent Advances, Next Generation Data Mining, Editors: H. Kargupta, J. Han, P.S. Yu, R. Motwanu and V. Kumar, Chapter 1, CRC Press.

[25] Roddick, J.F., Hornsby, K., and Spiliopoulou, M. (2001), An updated bibiliography on temporal, spatial, and spatio-temporal data mining research, *Lecture Notes in Computer Science*, 2007, 147–163.

[26] Celik, M., Shekhar, S., Rogers, J., and Shine, J. (2008), Mixed-Drove Spatiotemporal Co-Occurrence Pattern Mining, *IEEE Transactions on Knowledge and Data Engineering*, 20(10), 1322–1335.

[27] Bruno, A. and Appice, A. (2011), Marine Traffic Engineering through Relational Data Mining, Proceedings of The Workshop on Mining Complex Patterns, MCP 2011.

[28] Oo, K.M.S., Shi, C., and Weintrit, A. (2004), Clustering Analysis and Identification of Marine Traffic Congested Zones at Wusongkou, Shangai.

[29] Tang, C. and Shao, Z. (2009), Data Mining Platform Based on AIS Data, Proceedings of the International Conference on Transportation Engineering 2009 (ICTE 2009).

[30] Feixiang, Z. (2011), Mining ship spatial trajectory patterns from AIS database for maritime surveillance, Proceedings of 2nd Intl. conference on Emergency Management and Management Science, pp. 772–775.

[31] Zheng, B., Chen, J., Xia, S., and Jin, Y. (2008), Data Analyisis of Vessel Traffic Flow Using Clustering ALgorithms, Proceedings of the Int'l Conference on Intelligent Computation Technology and Automation, pp. 243–246.

[32] Tsou, M. C. (2010), Discovering Knowledge from AIS Database for Application in VTS, *Navigation*, 63(3), 449–469.

[33] Ou, Z. and Zhu, J. (2008), AIS Database Powered by GIS Technology for Maritime Safety and Security, *Navigation*, 61, 655–665.

[34] SKYbrary: NASA Data Mining Algorithms (online), `http://www.skybrary.aero/index.php/NASA_Data_Mining_Algorithms`.

[35] The Maritime Safety Office (online), `http://msi.nga.mil/NGAPortal/MSI.portal`.

[36] Tozicka, J., Rovatsos, M., Pehoucek, M., and Urban, S. (2008), MALEF: Framework for distributed machine learning and data mining, *Intelligent Information and Database Systems*, 2, 6–24.

[37] Mikut, R. and Reischl, M. (2011), Data mining tools, *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, 1(5), 431–443.

[38] (2012), CRAN Task View: Analysis of Spatial Data (online), `http://cran.r-project.org/web/views/Spatial.html` (Access Date: 2012).

[39] Klimberg, R. K. and Miori, V. (2010), Back in Business, *OR/MS Today*, 37(5).

[40] M., Golfarelli (2009), Open Source BI Platforms: A Functional and Architectural Comparison.

[41] Badard, T., Dube, E., Diallo, B., Mathieu, J., and Ouattara, M. (2009), Open source geospatial BI in action (online), `http://geosoa.scg.ulaval.ca/~badard/geocamp2009-open_source_geospatial_bi_in_action-tbadard.pdf` (Access Date: 2013).

[42] (2012), BI components of SpagoBI (online), SpagoBI, `http://www.spagoworld.org/xwiki/bin/view/SpagoBI/BIComponents` (Access Date: 2012).

[43] Analytics, Rexer (2012), 5th Annual Data Miner Survey - 2011 Survey Summary Report. Upon request at `http://www.rexeranalytics.com/Data-Miner-Survey-Results-2011.html`.

[44] (2012), What Analytics, Data mining, Big Data software you used in the past 12 months for a real project? (online), http://www.kdnuggets.com/polls/2012/analytics-data-mining-big-data-software.html (Access Date: 2012).

[45] Knorr, Edwin M. and Ng, Raymond T. (1998), Algorithms for Mining Distance-Based Outliers in Large Datasets, pp. 392–403.

[46] Deconstructing Adult Zebrafish Behavior with Swim Trace Visualizations (online), http://www.kaluefflab.com/publications.html.

[47] Valero-Mora, Pedro M. and Ledesma, Ruben (2012), Graphical User Interfaces for R, *Journal of Statistical Software*, 49(1), 1–8.

[48] Hiebeler, David (2011), MATLAB / R Reference. http://www.math.umaine.edu/~hiebeler/comp/matlabR.pdf.

[49] Raymond, Eric S. (2011), AIVDM/AIVDO protocol decoding, version 1.32. http://gpsd.berlios.de/AIVDM.html.

This page intentionally left blank.

Annex A: RapidMiner appendix

A.1 Invalid Observation Detection

```
<?xml version="1.0" encoding="UTF-8" standalone="no"?>
<process version="5.3.000">
  <context>
    <input/>
    <output/>
    <macros/>
  </context>
  <operator activated="true" class="process" compatibility="5.3.000"
  expanded="true" name="Process">
    <process expanded="true" height="251" width="547">
      <operator activated="true" class="read_csv" compatibility="5.3.000"
      expanded="true" height="60" name="Read CSV" width="90" x="45" y="30">
        <parameter key="csv_file"
        value="/home/mhadzagi/Telechargements/msari_error_total.csv"/>
        <parameter key="column_separators" value=","/>
        <parameter key="date_format" value="yyyy-MM-dd HH:mm:ss"/>
        <parameter key="first_row_as_names" value="false"/>
        <list key="annotations">
          <parameter key="0" value="Name"/>
        </list>
        <parameter key="encoding" value="UTF-8"/>
        <list key="data_set_meta_data_information">
          <parameter key="0" value="report_id.true.nominal.attribute"/>
          <parameter key="1" value="mmsi.true.nominal.attribute"/>
          <parameter key="2" value="report_timestamp.true.date_time.attribute"/>
          <parameter key="3" value="latitude.true.real.attribute"/>
          <parameter key="4" value="longitude.true.real.attribute"/>
          <parameter key="5" value="quality.true.integer.attribute"/>
          <parameter key="6" value="message_type.false.integer.attribute"/>
          <parameter key="7" value="callsign.false.attribute_value.attribute"/>
          <parameter key="8" value="imo_number.false.attribute_value.attribute"/>
          <parameter key="9" value="name.false.attribute_value.attribute"/>
          <parameter key="10" value="ship_type.false.attribute_value.attribute"/>
          <parameter key="11" value="ais_source_id.false.binominal.attribute"/>
          <parameter key="12"
          value="dimension_to_bow.false.attribute_value.attribute"/>
          <parameter key="13"
          value="dimension_to_port.false.attribute_value.attribute"/>
```

```
          <parameter key="14"
          value="dimension_to_starboard.false.attribute_value.attribute"/>
          <parameter key="15"
          value="dimension_to_stern.false.attribute_value.attribute"/>
          <parameter key="16" value="eta_month.false.attribute_value.attribute"/>
          <parameter key="17" value="eta_day.false.attribute_value.attribute"/>
          <parameter key="18" value="eta_hour.false.attribute_value.attribute"/>
          <parameter key="19" value="eta_minute.false.attribute_value.attribute"/>
          <parameter key="20" value="draught.false.attribute_value.attribute"/>
          <parameter key="21"
          value="destination_location.false.attribute_value.attribute"/>
          <parameter key="22" value="navigational_status.false.integer.attribute"/>
          <parameter key="23" value="rate_of_turn.true.integer.attribute"/>
          <parameter key="24" value="speed.true.real.attribute"/>
          <parameter key="25" value="course.true.real.attribute"/>
          <parameter key="26" value="heading.true.integer.attribute"/>
        </list>
      </operator>
      <operator activated="true" class="filter_examples" compatibility="5.3.000"
      expanded="true" height="76" name="Filter Examples" width="90" x="246" y="30">
        <parameter key="condition_class" value="attribute_value_filter"/>
        <parameter key="parameter_string" value="latitude=91&amp;amp;amp;
        longitude =181"/>
      </operator>
      <connect from_op="Read CSV" from_port="output" to_op="Filter Examples"
      to_port="example set input"/>
      <connect from_op="Filter Examples" from_port="example set output"
      to_port="result 1"/>
      <portSpacing port="source_input 1" spacing="0"/>
      <portSpacing port="sink_result 1" spacing="0"/>
      <portSpacing port="sink_result 2" spacing="0"/>
    </process>
  </operator>
</process>
```

A.2 Association Rule Mining using FP-Growth Algorithm

```
<?xml version="1.0" encoding="UTF-8" standalone="no"?>
<process version="5.3.000">
  <context>
```

```
  <input/>
  <output/>
  <macros/>
</context>
<operator activated="true" class="process" compatibility="5.3.000"
expanded="true" name="Process">
  <process expanded="true" height="633" width="567">
    <operator activated="true" class="read_database" compatibility="5.3.000"
    expanded="true" height="60" name="Read Database" width="90" x="45" y="30">
      <parameter key="connection" value="Localhost"/>
      <parameter key="define_query" value="table name"/>
      <parameter key="table_name" value="Contact_info"/>
      <enumeration key="parameters"/>
      <parameter key="datamanagement" value="sparse_map"/>
    </operator>
    <operator activated="true" class="select_attributes" compatibility="5.3.000"
    expanded="true" height="76" name="Select Attributes" width="90"
    x="112" y="120">
      <parameter key="attribute_filter_type" value="subset"/>
      <parameter key="attributes" value="|report_timestamp|port|mmsi|cell_id"/>
    </operator>
    <operator activated="true" class="numerical_to_polynominal"
    compatibility="5.3.000" expanded="true" height="76"
    name="Numerical to Polynominal" width="90" x="179" y="30">
      <parameter key="attribute_filter_type" value="single"/>
      <parameter key="attribute" value="mmsi"/>
    </operator>
    <operator activated="true" class="loop_values" compatibility="5.3.000"
    expanded="true" height="76" name="Loop Values" width="90" x="313" y="30">
      <parameter key="attribute" value="mmsi"/>
      <parameter key="iteration_macro" value="mmsi_id"/>
      <parameter key="parallelize_iteration" value="true"/>
      <process expanded="true" height="651" width="585">
        <operator activated="true" class="filter_examples"
        compatibility="5.3.000" expanded="true" height="76"
        name="Filter Examples" width="90"
        x="45" y="30">
          <parameter key="condition_class" value="attribute_value_filter"/>
          <parameter key="parameter_string" value="mmsi=%{mmsi_id}"/>
        </operator>
        <operator activated="true" class="remove_duplicates"
        compatibility="5.3.000" expanded="true" height="76"
```

```
name="Remove Duplicates" width="90" x="179" y="30">
  <parameter key="attribute_filter_type" value="subset"/>
  <parameter key="attributes" value="cell_id|mmsi"/>
</operator>
<operator activated="true" class="set_role"
compatibility="5.3.000" expanded="true" height="76"
name="Set Role" width="90" x="313" y="30">
  <parameter key="name" value="port"/>
  <parameter key="target_role" value="label"/>
  <list key="set_additional_roles"/>
</operator>
<operator activated="true" class="set_role" compatibility="5.3.000"
expanded="true" height="76" name="Set Role (2)" width="90"
x="45" y="165">
  <parameter key="name" value="mmsi"/>
  <parameter key="target_role" value="id"/>
  <list key="set_additional_roles"/>
</operator>
<operator activated="true" class="nominal_to_binominal"
compatibility="5.3.000" expanded="true" height="94"
name="Nominal to Binominal" width="90" x="179" y="165">
  <parameter key="attribute_filter_type" value="single"/>
  <parameter key="attribute" value="cell_id"/>
  <parameter key="include_special_attributes" value="true"/>
</operator>
<operator activated="true" class="fp_growth"
compatibility="5.3.000" expanded="true" height="76"
name="FP-Growth" width="90" x="179" y="300">
  <parameter key="find_min_number_of_itemsets" value="false"/>
  <parameter key="min_number_of_itemsets" value="5"/>
  <parameter key="min_support" value="0.01"/>
</operator>
<operator activated="true" class="create_association_rules"
compatibility="5.3.000" expanded="true" height="76"
name="Create Association Rules" width="90" x="313" y="300">
  <parameter key="min_confidence" value="0.001"/>
</operator>
<connect from_port="example set" to_op="Filter Examples"
to_port="example set input"/>
<connect from_op="Filter Examples" from_port="example set output"
to_op="Remove Duplicates" to_port="example set input"/>
<connect from_op="Remove Duplicates" from_port="example set output"
```

```
          to_op="Set Role" to_port="example set input"/>
          <connect from_op="Set Role" from_port="example set output"
          to_op="Set Role (2)" to_port="example set input"/>
          <connect from_op="Set Role (2)" from_port="example set output"
          to_op="Nominal to Binominal" to_port="example set input"/>
          <connect from_op="Nominal to Binominal" from_port="example set output"
          to_op="FP-Growth" to_port="example set"/>
          <connect from_op="FP-Growth" from_port="frequent sets"
          to_op="Create Association Rules" to_port="item sets"/>
          <connect from_op="Create Association Rules" from_port="rules"
          to_port="out 1"/>
          <portSpacing port="source_example set" spacing="0"/>
          <portSpacing port="sink_out 1" spacing="0"/>
          <portSpacing port="sink_out 2" spacing="0"/>
        </process>
      </operator>
      <connect from_op="Read Database" from_port="output" to_op="Select Attributes"
      to_port="example set input"/>
      <connect from_op="Select Attributes" from_port="example set output"
      to_op="Numerical to Polynominal" to_port="example set input"/>
      <connect from_op="Numerical to Polynominal" from_port="example set output"
      to_op="Loop Values" to_port="example set"/>
      <connect from_op="Loop Values" from_port="out 1" to_port="result 1"/>
      <portSpacing port="source_input 1" spacing="0"/>
      <portSpacing port="sink_result 1" spacing="0"/>
      <portSpacing port="sink_result 2" spacing="0"/>
    </process>
  </operator>
</process>
```

This page intentionally left blank.

Annex B: SQL Query for Creating Routes from Contact_info DB

```
INSERT INTO Routes.Contact_routes
select a.mmsi, a.port, b.port, a.report_timestamp, b.report_timestamp
FROM Routes.Contact_info a, Routes.Contact_info b
WHERE
a.mmsi = b.mmsi
AND a.report_timestamp < b.report_timestamp
AND a.port != b.port
GROUP BY a.mmsi%
DELETE FROM Routes.Contact_routes WHERE porta != null AND mmsi != null
SELECT * FROM Routes.Contact_routes
%
```

This page intentionally left blank.

Annex C: R Commands For Scenarios Analysis

C.1 R Commands for Mining Invalid Observations

Import packages:

```
> library(DBI)
> library(RPostgreSQL)
```

Open database connection:

```
> drv <- dbDriver("PostgreSQL")
> con <- dbConnect(drv, dbname = "msari_mo", host='192.000.00.000',user='postgres',
  password='dummy',port='5432')
```

Query the database for reports with out-of-range position, MMSI and attributes:

```
> sqlQuery = "SELECT   report_id,mmsi,report_timestamp,latitude,longitude,quality,
 message_type,callsign,imo_number,name,ship_type,ais_source_id,dimension_to_bow,
 dimension_to_port,dimension_to_starboard,dimension_to_stern,eta_month,eta_day,
 eta_hour,eta_minute,draught,destination_location,navigational_status,
 rate_of_turn,speed,course,heading
 FROM filter_details('2011-11-03','2011-11-10',NULL,NULL,NULL,NULL,NULL,
 'exact earth', 'ais')
 WHERE ( (quality & 2) = 2 OR (quality & 4) = 4 OR (quality & 256) =256)
 AND entity_type='VESSEL'
 ORDER BY mmsi,report_timestamp,message_type"

> rs <- dbSendQuery(con, statement = paste(sqlQuery))
> msari <- fetch(rs, n = -1)
```

The MSARI filter function returns all values as string. In order to verify if the values are out-of-range, some values have to be converted in numerical format:

```
> msari$draught <-as.numeric(msari$draught)
> msari$dimension_to_starboard <-as.numeric(msari$dimension_to_starboard)
> msari$dimension_to_stern <-as.numeric(msari$dimension_to_stern)
> msari$dimension_to_bow <-as.numeric(msari$dimension_to_bow)
```

```
> msari$dimension_to_port <-as.numeric(msari$dimension_to_port)
> msari$rate_of_turn <-as.numeric(msari$rate_of_turn)
> msari$navigational_status <-as.numeric(msari$navigational_status)
> msari$eta_minute <-as.numeric(msari$eta_minute)
> msari$eta_hour <-as.numeric(msari$eta_hour)
> msari$eta_day <-as.numeric(msari$eta_day)
> msari$eta_month <-as.numeric(msari$eta_month)
> msari$heading <-as.numeric(msari$heading)
> msari$speed <-as.numeric(msari$speed)
> msari$course <-as.numeric(msari$course)
```

Note that the scenario was also executed with a CSV file as input. This CSV file contains the result of the query above. In that case, the data is imported with:

```
> msari <- read.csv(file="C:\\Users\\...\\msari_error_total.csv",head=TRUE,sep=",")
```

The next step is to transform some of the values of the data set `msari` to 0 or 1, depending if they are valid or not:

```
> msari$longitude <- ifelse(msari$longitude > 181 | msari$longitude <(-181), 1, 0)
> msari$latitude <- ifelse(msari$latitude > 180 & msari$latitude <(-90), 1, 0)
> msari$navigational_status<- ifelse(msari$navigational_status > 15
  | msari$navigational_status <(0), 1, 0)
> msari$rate_of_turn <- ifelse(msari$rate_of_turn > 1000
  | msari$rate_of_turn <(0), 1, 0)
> msari$speed <- ifelse(msari$speed > 102.3 | msari$speed <(0), 1, 0)
> msari$course <- ifelse(msari$course > 360 | msari$course <(0), 1, 0)
> msari$heading <- ifelse((msari$heading > 359 | msari$heading <(0))
  | msari$heading==511, 1, 0)
> msari$ship_type <- ifelse((msari$ship_type> 99 | msari$ship_type <(0)) , 1, 0)
> msari$mmsi <- ifelse((nchar(msari$mmsi)!=9) , 1, 0)
> msari$imo_number <- ifelse((nchar(msari$imo_number)!=7 &
  !is.na(msari$imo_number)) , 1, 0)
> msari$dimension_to_bow <- ifelse((msari$dimension_to_bow> 511
  | msari$dimension_to_bow <(0))& !is.na(msari$dimension_to_bow) , 1, 0)
> msari$dimension_to_stern <- ifelse((msari$dimension_to_stern> 511
  | msari$dimension_to_stern <(0))& !is.na(msari$dimension_to_stern) , 1, 0)
> msari$dimension_to_port <- ifelse((msari$dimension_to_port> 63
  | msari$dimension_to_port <(0))& !is.na(msari$dimension_to_port) , 1, 0)
> msari$dimension_to_starboard <- ifelse((msari$dimension_to_starboard> 63
  | msari$dimension_to_starboard <(0))& !is.na(msari$dimension_to_starboard) , 1, 0)
```

```
> msari$eta_hour <- ifelse((msari$eta_hour> 24 | msari$eta_hour <(0))
  & !is.na(msari$eta_hour) , 1, 0)
> msari$eta_month <- ifelse((msari$eta_month> 12 | msari$eta_month <(0))
  & !is.na(msari$eta_month) , 1, 0)
> msari$eta_day <- ifelse((msari$eta_day> 31 | msari$eta_day <(0))&
  !is.na(msari$eta_day) , 1, 0)
> msari$eta_minute <- ifelse((msari$eta_minute> 60 | msari$eta_minute <(0))
  & !is.na(msari$eta_minute) , 1, 0)
```

The frequency $\mathcal{F}(A|B)$ is computed with the following user-defined function:

```
> frequence <- function(x,y) { return(100*sum(x*y)/sum(y)) }
```

For instance, $\mathcal{F}(ETAmonth|Shiptype)$ is computed with:

```
> frequence(msari$eta_month,msari$ship_type)
```

C.2 R Commands for Mining Ship Trajectories

Only MMSI and cell IDs are required for the analysis. The others columns are removed. Also, all duplicated rows are removed. A duplicate is defined by a ship reported in the same cell for more than one consecutive time period. Only the unique cells visited by each ship are required for the analysis.

```
> reports_tmp <- reports[c(-2,-3,-5)]
> reports_clean <- unique(reports_tmp)
```

C.2.1 Sub-Case 1

The first command gets all ships that visited ports of Quebec (cell ID 67798) or Montreal (cell ID 66461). The second one lists ships (identified by MMSI) that went to both ports and visited port of Quebec last:

```
> visitors <- subset(reports_clean,reports_clean$cell_id=="66461"
  | reports_clean$cell_id=="67798")
> visitors[ which(visitors$cell_id=='67798' & duplicated(visitors$mmsi)),]$mmsi
```

C.2.2 Sub-Case 2

The first step is to remove all ships with only one contact:

```
> d <- subset(b, Freq >1,select=c(Var1))
> contacts_2<-reports_clean[reports_clean$mmsi %in% d$Var1,]
```

Then, the data frame has to be transformed into an incidence matrix and then an object of class transactions:

```
> library(arules)
> matrix_contacts_2 <- as.matrix(unclass(table(contacts_2)))
> dimnames(matrix_contacts_2) <- list(as.character(unique(contacts_2$mmsi)),
  as.character(unique(contacts_2$cell_id)))
> transac <-  as(matrix_contacts_2, "transactions")
```

The association algorithm `apriori` can be applied on the transactions object. The support and confidence are set to low values in order to get a lot of rules.

```
> rules_2 <- apriori(transac,parameter=list(support=0.01,confidence=0.1,maxlen=2))
```

The port data is imported and a subset of rules implying only ports is created.

```
> port <- read.csv(file="C:\\Users\\...\\port_to_cell.csv",head=TRUE,sep=",")
> port_cell <-as.character(port$cell_id)

> rulesWithPorts_2 <- subset(rules, lhs %in% port_cell & rhs %in% port_cell )
> inspect(rulesWithPorts_2)
```

Results can be visualized with the `arulesViz` package. The following command allows users to visualize rules as a graph:

```
> library(arulesViz)
> plot(rulesWithPorts_2, method="graph",measure="confidence",
  control=list(main="Rules linking ports ID",alpha=1,arrowSize=0.8))
```

List of symbols/abbreviations/acronyms/initialisms

ADS-B	Automatic Dependent Surveillance-Broadcast
AIS	Automatic Identification System
ANZ	Australia and New Zealand
API	Application Programming Interface
ARFF	Attribute-Relation File Format
BI	Business Intelligence
CRAN	Comprehensive R Archive Network
CSV	Comma-Separated Values
DB	Database
DBMS	Database Management System
DBSCAN	Density-Based Spatial Clustering of Applications with Noise
DM	Data Mining
DRDC	Defence Research and Development Canada
EML	Ecological Metadata Language
EOS	Earth Observing System
ETA	Estimated Time of Arrival
FTP	File Transfer Protocol
GDAL	Geospatial Data Abstraction Library
GDPM	Geographic Data Pre-processing Module
GIS	Geographic Information System
GPS	Global Positioning System
GPW	Global Position Warehouse
GUI	Graphical User Interface
HTML	HyperText Markup Language
HTTP	Hypertext Transfer Protocol
ID	Identification Number
IDE	Integrated Development Environment
IMO	International Maritime Organization
IP	Internet Protocol
JDBC	Java Database Connectivity

KDD	Knowledge Discovery in Databases
KPI	Key Performance Indicators
MARLANT	Maritime Forces Atlantic
MCP	Marine Community Profile
MDCOPs	Mixed-Drove Spatio-temporal Co-Occurrence Patterns
MDX	The MultiDimensional eXpressions
MEF	Metadata Exchange Format
MIS	Maritime Information Support
MMC	Multipurpose Marine Cadastre
MMSI	Maritime Mobile Service Identity
MSA	Maritime Situational Awareness
MSARI	Maritime Situational Awareness Research Infrastructure
MSOC	Marine Security Operations Centres
NASA	National Aeronautics and Space Administration
NGA	National Geospatial-Intelligence Agency
NLP	Natural Language Processing
NOAA	National Oceanic and Atmospheric Administration
NODC	National Oceanographic Data Center
OLAP	OnLine Analytical Processing
OS	Operating System
RDBMS	Relational Database Management System
RJOC	Regional Joint Operations Center
ROC	Receiver Operating Characteristic
SDM	Spatial Data Mining
SQL	Structured Query Language
SVM	Support Vector Machine
URL	Uniform Resource Locator
US	United States
UUID	Universally Unique Identifier
WEKA	Waikato Environment for Knowledge Analysis
WMS	Web Map Service
WFS	Web Feature Service
XML	eXtensible Markup Language